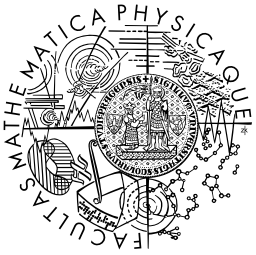


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Evgeniya Knyazeva

Komunity a jejich detekce v sociálních sítích

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: doc. RNDr. Iveta Mrázová, CSc.

Studijní program: Informatika (B1801)

Studijní obor: Obecná informatika

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ....

Podpis autora

Tímto bych ráda poděkovala vedoucí mé práce doc. RNDr. Ivetě Mrázové, CSc. za podnětné připomínky a rady. Dále bych ráda poděkovala svému příteli a rodině za morální podporu a pochopení.

Název práce: Komunity a jejich detekce v sociálních sítích

Autor: Evgeniya Knyazeva

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: doc. RNDr. Iveta Mrázová, CSc., Katedra teoretické informatiky a matematické logiky

Abstrakt: Analýza sociálních sítí se dá využít ke zkoumání struktury společnosti, jejího vývoje a chování lidí v ní. V této bakalářské práci jsme se zaměřili především na detekci komunit v sociálních sítích. Představili jsme základní způsoby analýzy významných vrcholů v síti. Rozebrali jsme několik různých přístupů k detekci komunit – detekce hierarchické struktury sítě, překrývajících se komunit a komunit v dynamických sítích. Představili jsme funkci zvanou modularita a její využití při detekci komunit. Práce se následně blíže zaměřuje na implementaci několika algoritmů (Lovaňský algoritmus, algoritmus klastrování pomocí hran, SCAN, DSCAN) v jazyce Python s použitím knihovny NetworkX. Tyto algoritmy jsme následně aplikovali na e-mailovou síť zaměstnanců firmy Enron. V práci jsme se pokusili odhalit komunity v této síti a vliv událostí, které zasáhly firmu, na jejich strukturu.

Klíčová slova: dobývání znalostí, sociální sítě, detekce komunit, reprezentace znalostí

Title: Communities and their Detection in Social Networks

Author: Evgeniya Knyazeva

Department: Name of the department

Supervisor: doc. RNDr. Iveta Mrázová, CSc., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Social networks analysis can be used to study the society's structure, its development and the people's behavior. In this bachelor thesis, we focus mainly on detecting communities in social networks. We introduce the basic methods for significant nodes analysis. We discuss several different approaches to detecting communities, namely detecting hierarchical network structure, overlapping communities and communities in dynamic networks. We introduce a function called 'modularity' and describe its use in community detection. The work then focuses on implementing several algorithms (Louvain algorithm, Link clustering algorithm, SCAN, DSCAN) in Python using the NetworkX library. We then applied the algorithms to Enron employees e-mail network and attempted to reveal communities in it and describe how their structure was affected by major events in the company.

Keywords: data mining, social networks, community detection, knowledge representation

Obsah

Úvod

Internetové sociální sítě zasahují dnes již do života každého z nás. Informace o uživatelích a jejich vzájemné vztahy mohou být užitečné při analýze chování jednotlivců i společnosti jako celku. V této práci se zaměříme především na analýzu struktury sítí, konkrétně na detekci jejich hustě propojených částí, kterým v kontextu sociálních sítí říkáme komunity.

Jako první představíme definici sociálních sítí (Kapitola 1). Popíšeme jejich základní vlastnosti (Podkapitola 1.1) a možné definice komunit (Podkapitola 1.2). Následně zanalyzujeme způsoby reprezentace komunit a které jejich konkrétní definice využijeme v této práci (Podkapitola 1.4).

Uvedeme několik metod pro analýzu sociálních sítí (Kapitola 2). Jako první si představíme způsoby detekce důležitých vrcholů (Podkapitola 2.1). Jako příklad takového vrcholu lze uvést vrchol s nejvyšším stupněm nebo artikulaci [1]. V podkapitole 2.2 se podíváme na různé přístupy k detekci komunit: hledání hierarchické struktury a překrývajících se komunit a analýzu komunit v dynamických sítích. Kromě toho představíme koncept modularity, kterou využijeme nejenom k detekci komunit, ale i k vyhodnocování kvality výsledných rozdělení vrcholů na komunity. V podkapitole 2.3 se blíže podíváme na několik algoritmů, které pokrývají oblasti představené v druhé části.

Vybranými algoritmy jsou Lovaňský algoritmus, algoritmus klastrování pomocí hran, SCAN a jeho dynamická verze DSCAN (Podkapitole 2.3). Implementovali jsme je v jazyce Python s použitím knihovny NetworkX. Pomocí těchto algoritmů budeme zkoumat strukturu komunit v sítích, které jsme vytvořili z e-mailové komunikace zaměstnanců společnosti Enron z několika let předcházejících jejímu zániku (Kapitola 3). Celkem jsme ze zpráv zkonstruovali čtyři statické sítě a jednu dynamickou, která je tvořená několika vzorky, z nichž každý reprezentuje jedno čtvrtletí sledovaného období.

Každý z vybraných algoritmů jsme použili na prozkoumání specifických vlastností zkonstruovaných sítí. Pomocí Lovaňského algoritmu (Podkapitola 2.3.1) jsme se pokusili odhalit hierarchickou strukturu. Algoritmus jsme navíc použili na analýzu vlivu orientace a ohodnocení spojení na strukturu komunit. Algoritmus SCAN (Podkapitola 2.3.2) se kromě komunit zaměřuje na detekci rozcestníků – vrcholů, které spojují různé komunity – a odlehlých vrcholů, které nepatří do žádné komunity a zároveň nejsou rozcestníky. Použili jsme ho tedy na odhalení vlivu vrcholů v rámci sítě. Algoritmem klastrování pomocí hran (Podkapitola 2.3.3) jsme mohli zanalyzovat výskyt překrývajících se komunit. Algoritmus DSCAN (Podkapitola 2.3.4) jsme využili pro analýzu změn v struktuře komunit během sledovaného období.

U každého rozdělení na komunity jsme se kromě struktury zabývali také reprezentativností komunit, tj. zkoumali jsme, jestli námi detekované skupiny mají základ v reálném světě. K těmto účelům jsme využili především klíčová slova z předmětů zpráv posílaných v rámci jednotlivých komunit. Podařilo se nám nalézt několik skupin uživatelů, kteří by mohli tvořit komunity ve skutečném světě, a tyto skupiny se vyskytovaly jako základy komunit napříč všemi rozděleními nezávisle na použitém algoritmu (Podkapitola 4.2.1).

[1] Artikulace je vrchol, po jehož odebrání se v grafu zvýší počet komponent souvislosti.

Nakonec jsme ukázali, že zkoumaná síť nemá významnou hierarchickou strukturu a že váhy spojení a jejich orientace mají vliv na detekované komunity (Podkapitola 4.2.4). Podařilo se nám nalézt několik významných jedinců, kteří tvoří důležité komunikační body mezi různými komunitami, a odhalit několik různých druhů překrývajících se komunit (Podkapitola 4.4.2). Také se potvrdila naše hypotéza o vlivu zániku firmy na strukturu komunit – komunity se zmenšovaly, rozpadaly a zanikaly (Podkapitola 4.5.2).

1. Sociální sítě a komunity

Pod pojmem sociální síť si většina lidí představí internetové sociální sítě jako Facebook, Instagram, či Twitter. Za sociální síť ale můžeme považovat jakýkoliv sociální konstrukt, kde mezi prvky probíhá nějaký druh interakce. Jako příklad si můžeme uvést síť tvořenou žáky jednoho gymnázia a vztahy definovat jako známosti mezi nimi. V jiném příkladu můžeme vzít fotbalové týmy a interakce definovat třeba počtem hráčů, kteří během určitého časového období přestoupili z jednoho týmu do druhého.

Pomocí analýzy vztahů mezi prvky v síti dokážeme například detekovat jedince, kteří jsou pro danou síť významní. Obvykle nás zajímá počet spojení daného jedince se zbytkem sítě, nebo množství informací, které přes daného jedince prochází. Pokud bychom opět použili příklad se studenty gymnázia, tak za významné jedince v síti můžeme považovat určitě takové, kteří mají nejvíce známostí z jiné než vlastní třídy, nebo například třídní zástupce, kteří zprostředkovávají komunikaci třídy jako celku s učiteli a vedením školy.

Sociální sítě se vyznačují tím, že se v nich často tvoří shluky. Proto kromě vyhledávání významných jedinců lze v sítích vyhledávat i zmíněné shluky, kterým říkáme komunity. Komunity obvykle sdružují jedince s podobnými vlastnostmi a vyznačují se tím, že spojení mezi jedinci v rámci jedné komunity jsou obvykle velmi hustá. V příkladu se studenty gymnázia se jako komunity nabízí jednotlivé třídy nebo spolky studentů se stejnými zájmy, jako například dramatický kroužek, sbor, fotbalový tým...

Sociální síť je tedy množinou jedinců, mezi kterými máme nějakým způsobem definované vztahy. V sítích můžeme detekovat významné jedince a jejich vliv na ostatní jedince v síti, nebo můžeme zkoumat výskyt komunit v síti a vlastnosti těchto komunit. V textu budeme používat také pojem dynamické sociální sítě. Představme si pod ním síť, která se mění v čase, tj. přibývají a ubývají v ní vrcholy a hrany.

V této kapitole si krátce představíme typické vlastnosti sociálních sítí, možnosti jejich reprezentace, způsoby definice komunit a jakou tyto definice hrají roli při detekci komunit. Na závěr si představíme sociální síť, se kterou budeme pracovat v této práci.

1.1 Vlastnosti sociálních sítí a možnosti jejich reprezentace

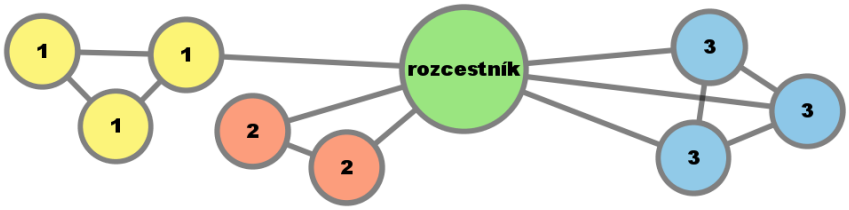
Sociální síť obvykle reprezentujeme jako graf, kde vrcholy tvoří jedinci v síti a hrany definují vztahy mezi nimi. Pokud jsou vztahy definovány pouze jednostranně, výsledný graf bude orientovaný. K jednotlivým vrcholům a hranám lze přidat seznamy vlastností, nebo obsah, který s daným uživatelem (nebo vztahem) souvisí. Od této chvíle budeme jedincům v síti říkat vrcholy (nebo uzly) a spojením mezi jedinci hrany.

Nyní se podívejme na některé typické vlastnosti sociálních sítí [1]. Většinou jsou sítě tvořeny hustě propojenými komunitami, které jsou navzájem spojeny malým počtem hran. Kromě zmíněné tendence tvořit shluky můžeme v sítích

často najít významné vrcholy, kterým říkáme *rozcestníky* (Obrázek 1.1). Jsou to obvykle vrcholy, které nejsou součástí žádné komunity a tvoří mezi nimi spojnice.

Definujme si vzdálenost mezi dvěma uzly jako počet jiných uzlů na nejkratší cestě mezi uzly zkoumanými. Další vlastnost popisuje fenomén „svět je malý“, který nám říká, že vzdálenost mezi každými dvěma uzly v síti je malá.

Pro dynamické sítě potom platí také následující dvě vlastnosti. Čím více má vrchol sousedů, tím jednodušeji bude schopný navázat nová spojení, tj. nově přidané vrcholy budou spíše napojeny na nějaký vrchol s vyšším stupněm a nově přidané hrany budou opět propojovat vrcholy s vyšším stupněm se zbytkem sítě. Druhá vlastnost se týká rozšiřování dynamické sítě. V průběhu času totiž vzniká spíše více nových hran než nových vrcholů. Tímto dochází k postupnému zahušťování sítě.



Obrázek 1.1: Ukázka rozcestníku spojujícího několik různých komunit. Každá komunita je označená vlastní barvou.

1.2 Definice komunit

Základní představa o komunitách se zakládá na následující hypotéze „Komunita je lokálně hustě propojený podgraf v síti.“ [2] Této hypotéze ale odpovídá více různých definic.

Pokud bychom uvážovali, že komunita je skupinou jednotlivců, kde se každý zná s každým, tak lze komunitu definovat jako úplný podgraf neboli kliku. Tato definice je ale v mnoha ohledech omezující. Výskyt klik je totiž v sítích poměrně vzácný a zavrhli bychom tak mnoho potenciálních komunit nevyhovujících této přísné definici.

Abychom zmírnili kritéria předchozí definice, zavedeme si lehkou toleranci, a to pomocí porovnání vstupního a výstupního stupně komunity [2].

Mějme libovolnou komunitu C a vezměme bod i z C. Označme k_i^{int} vnitřní stupeň vrcholu i a definujme ho jako počet sousedů z komunity C. Označme k_i^{ext} počet sousedů i mimo komunitu C. Čím menší je k_i^{ext}, tím více vyhovující je komunita C pro vrchol i. Čím více se ale k_i^{int} blíží nule, tím pravděpodobnější je, že pro vrchol existuje jiná, lépe vyhovující komunita.

Jako silnou komunitu pak definujeme komunitu C, kde pro každý vrchol $i \in C$ platí $k_i^{int} > k_i^{ext}$. Slabá komunita je potom taková, kde celkový vnitřní stupeň komunity je ostře větší než stupeň vnější: $\sum_{i \in C} k_i^{int} > \sum_{i \in C} k_i^{ext}$. Pozorujeme, že každá klika je silnou komunitou a každá silná komunita je zároveň slabou.

V algoritmech se pro sdružování vrcholů do komunit často využívá *podobnosti* vrcholů. Podobnost je definována jako funkce, která je obvykle závislá na množinách společných sousedů zkoumaných vrcholů, jejich stupních, nebo jiných spo-

lečných vlastnostech. Jako příklad takové funkce můžeme uvést strukturální podobnost (Vzorec 2.12) definovanou pro detekci komunit v algoritmu SCAN (Podkapitola 2.3.2).

1.3 Počet komunit v síti

Při detekci komunit nás bude zajímat také jejich počet v síti [2]. Pokud bychom chtěli síť rozdělit na předem pevně daný počet komunit, lze to provést opakovaným hledáním minimálního řezu. Problém můžeme dále dodefinovat tak, že se pokusíme najít řez takový, aby od sebe oddělené části měly přibližně stejnou velikost. Tímto způsobem můžeme iterativně rozdělovat graf na menší a menší části. Jsme ale dopředu omezeni jak počtem, tak velikostí výsledných komunit, což je i hlavní nevýhoda daného přístupu.

Většinou ale nechceme být při detekci komunit omezeni ani fixním počtem komunit, ani jejich velikostí. Jako první se nabízí vyzkoušet všechna možná rozdělení vrcholů na komunity. Počet možných rozdělení ale se zvětšující se velikostí grafu roste rychleji než exponenciálně. Proto pro detekci komunit použijeme definice popsané v podkapitole 1.2. Nebude již třeba zkoumat všechna možná rozdělení sítě na komunity, ale pouze specifické množiny vrcholů.

1.4 Sociální síť a komunity v této práci

V této práci se zaměříme na analýzu komunitní struktury v e-mailové síti tvořené zaměstnanci společnosti Enron (Kapitola 3). Jedinci jsou zaměstnanci firmy a vztahy mezi nimi definujeme počtem a druhem poslaných zpráv. Síť zachycuje komunikaci zaměstnanců z období těsně před zánikem firmy (od konce roku 1998 do poloviny roku 2002).

Síť proto budeme reprezentovat dvěma způsoby, jako statický a dynamický graf. Statický graf bude zachycovat souhrnnou komunikaci během celého sledovaného období. Dynamický graf bude tvořit několik nezávislých grafů, kde každý bude reprezentovat komunikaci mezi zaměstnanci v pevně daném časovém úseku. Vytvoříme dva druhy statických grafů – orientované a neorientované. Směr komunikace totiž může mít vliv jak na roli jedince v síti, tak i na strukturu komunit.

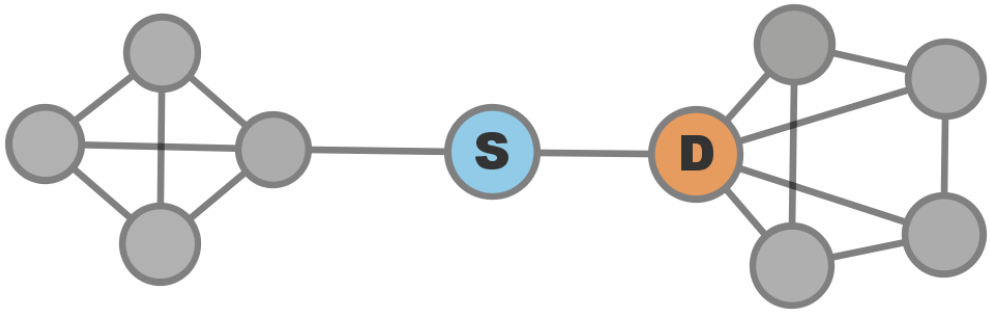
Předpokládáme, že komunity v této síti budou tvořeny skupinami zaměstnanců, kteří spolu pracovali na stejných projektech. Jako příklad můžeme uvést skupinu, která se zabývala prodejem a distribucí zemního plynu v Kalifornii (Podkapitole 4.4.1, část o reprezentativitě komunit). My se zaměříme na detekci komunit pomocí analýzy struktury sítě a předměty zpráv použijeme k ověření, že námi nalezené komunity mají základ ve skutečném světě.

2. Analýza sociálních sítí

Jak jsme zmínili v Kapitole 1, v sítích můžeme zkoumat významné vrcholy a strukturu komunit. V této kapitole si představíme několik možností detekce významných vrcholů a různé způsoby detekce komunit. Na závěr si blíže zanalyzujeme několik algoritmů, které jsme v rámci experimentů (Kapitola 4) aplikovali na zvolená data (Kapitola 3).

2.1 Detekce významných jedinců v síti

Při analýze sociálních sítí nás často zajímá, jestli se v ní vyskytují nějací významní jedinci. Můžou to být vrcholy s největším stupněm, nebo vrcholy, které mají v rámci vnitřní struktury sítě důležitou roli, například tvoří důležité spojení mezi různými částmi grafu (Obrázek 2.1). Proto pro detekci významných jedinců existuje mnoho přístupů a jejich použití závisí na tom, co přesně chceme zkoumat.



Obrázek 2.1: Ukázka významných vrcholů v grafu.
Vrchol **D** je významný hned ze dvou důvodu – má v grafu nejvyšší stupeň a spojuje všechny vrcholy pravé komunity. Přestože má **S** nejnižší stupeň v grafu, je také důležitým vrcholem, protože je jedinou spojnicí mezi levou a pravou komunitou.

2.1.1 Míry centrality

Centralita je mírou důležitosti vrcholu nebo hrany v síti. Existuje mnoho různých druhů centralit, níže uvedeme několik nejznámějších a nejpoužívanějších vrcholových centralit a jednu hranovou [3]. Definujeme je pro neorientované souvislé statické sítě. Ukázky vrcholů důležitých vzhledem k některým vybraným centralitám lze nalézt na Obrázku 2.2.

D-centralita (degree centrality, centralita měřená stupněm uzlu)

D-centralita je zjednodušeně stupeň vrcholu, který je definován jako počet sousedů daného vrcholu v grafu.[1]

[1] Pro orientované grafy zavádíme D-centralitu podle vstupních a výstupních stupňů vrcholů.

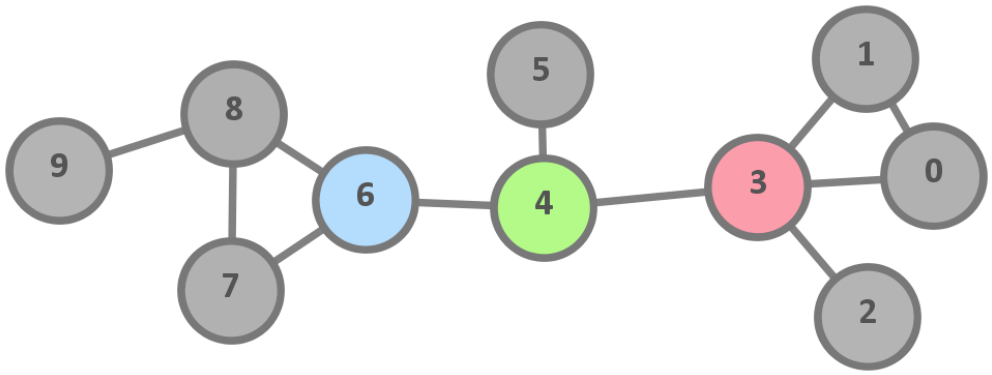
C-centralita (closeness centrality)

Nechť $AvgDist(i)$ je průměrná vzdálenost daného vrcholu od ostatních vrcholů v grafu, potom C-centralita je $1/AvgDist(i)$. Čím vyšší je hodnota centrality u vrcholu, tím jednodušší je dostat se z něj do všech částí grafu.

B-centralita (betweenness centrality)

Mějme graf (V, E) a vrchol $v \in V$. Definujme si $\sigma_{u,w}$ jako počet nejkratších cest mezi vrcholy u, w a $\sigma_{u,w}(v)$ jako počet nejkratších cest mezi u, w, které prochází vrcholem v. B-centralitu vrcholu v potom definujeme následujícím způsobem:

$$C_B(v) = \sum_{v \neq u \neq w \in V} \frac{\sigma_{u,w}(v)}{\sigma_{u,w}}. \tag{2.1}$$



Obrázek 2.2: Příklady B-, C- a D-centrality vrcholů v grafu.
V grafu jsou znázorněny některé významné vrcholy. Vrchol **3** má nejvyšší D-centralitu, a to 4. Vrchol **4** má nejvyšší B-centralitu (24) i normalizovanou C-centralitu (81/16). Dalšími vrcholy s vysokými hodnotami C- a B-centrality jsou **3** (C:81/18, B:20) a **6** (C:81/18, B:18).

B-centralita pro hrany (edge betweenness)

Je definována podobně jako B-centralita pro vrcholy [4]. Definujeme si $\sigma_{u,w}(e)$ jako počet nejkratších cest mezi u, w, které prochází hranou e. Potom

$$C_B(e) = \sum_{u \neq w \in V} \frac{\sigma_{u,w}(e)}{\sigma_{u,w}}. \tag{2.2}$$

je B-centralitou pro hranu e.

E-centralita (eigenvector centrality)

U této centrality nezáleží tolik na kvantitě spojení, které jeden vrchol má, jako na jejich kvalitě. Vrchol je tedy důležitý, pokud je napojený na jiné důležité vrcholy. Nechť je $A = (a_{i,j})$ matice sousednosti grafu. Potom

$$C_E(v) = \frac{1}{\lambda} \sum_{u \in V} a_{u,v} C_E(u) \tag{2.3}$$

je E-centralita vrcholu v, kde $\lambda \neq 0$ je konstanta. V maticové formě $\lambda C_E = C_E A$. Vektor C_E je vlastním vektorem matice sousednosti A a odpovídá mu vlastní číslo λ.

2.1.2 Algoritmy pro určování důležitosti vrcholů

Kromě výše zmíněných měr existuje i několik známých algoritmů na rozdělení vrcholů dle jejich důležitosti v grafu. Níže si popíšeme dva nejznámější a nejpoužívanější algoritmy [3].

HITS (Hyperlink-Induced Topic Search)

Tento algoritmus přiřazuje vrcholům dvě kategorie. *Rozcestníkem* nazveme vrchol, který má vysoký výstupní stupeň (odkazuje na mnoho jiných vrcholů). Jako *autoritu* definujeme vrchol, který má naopak vysoký vstupní stupeň (odkazuje na něj mnoho jiných vrcholů). Algoritmus tedy pro každý vrchol v spočítá dvě hodnoty, $H(v)$ – na kolik kvalitním je vrchol rozcestníkem, $A(v)$ – na kolik kvalitní je vrchol autoritou.

$$\begin{aligned} H(v) &= \sum_{u \in N(v)} A(u) \\ A(v) &= \sum_{u \in N(v)} H(u), \end{aligned} \tag{2.4}$$

kde $N(v)$ je množina sousedů vrcholu v.

PageRank

Je to iterativní algoritmus podobně jako HITS, liší se tím, že PageRank počítá pro každý vrchol pouze jednu hodnotu. Tato hodnota udává, s jakou pravděpodobností se při náhodné procházce grafem dá dostat do určitého vrcholu. Pro vrchol v je definován následně:

$$PR(v) = \frac{(1-d)}{N} + d \sum_{(u,v) \in E} \frac{PR(u)}{deg_{out}(u)}, \tag{2.5}$$

kde $deg_{out}(u)$ je výstupní stupeň vrcholu u, N počet všech vrcholů v grafu a d je tzv. *dumpingový*, nebo také *tlumicí faktor*.

2.2 Detekce komunit v síti

V první kapitole jsme si představili několik různých definic komunit (Podkapitola 1.2). V algoritmech na jejich detekci budeme používat hlavně poslední z definic, která byla založená na podobnosti vrcholů v komunitě.

V této části se zaměříme na několik různých přístupů, podle kterých lze algoritmy na detekci komunit klasifikovat [2]. Představíme si principy hierarchických algoritmů (Podkapitola 2.2.1), definujeme si kvalitativní funkci zvanou modularita (Podkapitola 2.2.2), krátce rozebereme možné způsoby detekce překrývajících se komunit (Podkapitola 2.2.3) a popíšeme principy detekce komunit v dynamických sítích (Podkapitola 2.2.4).

2.2.1 Hierarchické klastrování

Klastrování je proces hledání shluků v grafech. V kontextu sociálních sítí můžeme za klastrování považovat hledání rozdělení sítě na komunity.

Základem hierarchického klastrování je tzv. *podobnostní matice*, jejíž prvky x_{ij} indikují, jak moc jsou podobné vrcholy i a j. Podobnost vrcholů je většinou určena podobnostní funkcí, která je obvykle závislá na porovnání množin sousedů zkoumaných vrcholů. Hierarchické klastrování pak iterativně sdružuje do komunit vrcholy s vysokou podobností.

Existují dva hlavní přístupy – aglomerativní a divisivní. Aglomerativní algoritmy spojují do komunit vrcholy s vysokou podobností. Po spojení dvou komunit je třeba přepočítat podobnost nové komunity se zbylými. Spojování se opakuje, dokud všechny vrcholy nejsou součástí jedné velké komunity. Příkladem aglomerativního algoritmu je třeba algoritmus Ravaszové [5], nebo Lovaňský algoritmus [6].

Naopak divisivní algoritmy izolují komunity pomocí odstraňování spojení mezi vrcholy s nízkou podobností. Jako příklad může sloužit algoritmus Girvanové a Newmana (G-N) [4]. Jako podobnostní funkce v něm slouží hranová B-centralita (Vzorec 2.2). Chceme, aby pro dvojici vrcholů z různých komunit byla hodnota centrality vysoká, a naopak pro dvojice vrcholů v rámci jedné komunity byla hodnota centrality nízká. V případě G-N je v každé iteraci pro všechny hrany spočítana jejich B-centralita a odebrána hrana s nejvyšší hodnotou centrality.

Oba přístupy (aglomerativní a divisivní) generují hierarchický strom – dendrogram – popisující v jakém pořadí byly komunity spojovány dohromady. Pro získání rozdělení na komunity je třeba dendrogram v nějakém místě „rozříznout“. Volba řezu je ale libovolná. K rozhodování by mohly sloužit dodatečné znalosti o datech nebo kvalitativní funkce, které umí říct, jak dobré je dané rozdělení. V algoritmu G-N lze použít např. modularitu (Vzorec 2.6) [2].

Hierarchické algoritmy tedy umí odhalit nejenom rozdělení sítě na komunity, ale i případnou hierarchickou strukturu, která se za komunitami skrývá.

2.2.2 Modularita

V náhodně pospojovaných sítích se předpokládá, že spojení mezi vrcholy jsou rozdělena dle rovnoměrného rozdělení a jsou nezávislé na distribuci stupňů vrcholů v síti. Nevykazují tedy tendenci vytvářet shluky, a tudíž by se v nich neměly tvořit komunity.

Tohoto předpokladu můžeme použít pro porovnání hustoty spojení v algoritmem nalezené komunitě s hustotou spojení v náhodné síti a podle výsledků porovnání rozhodnout, zda se shluk vytvořil náhodně, nebo je součástí nějaké skutečné struktury. Zkoumání těchto odchylek od náhodné konfigurace nám umožňuje zavést pojem *modularity* [2], který nám poslouží nejen ke měření kvality získané komunity, ale také pomůže rozhodnout, které z několika rozdělení na komunity je lepší.

Uvažujme síť o N vrcholech a M hranách, její rozdělení do n_c komunit, kde každá komunita bude mít N_c vrcholů a L_c vnitřních hran. Pokud je L_c větší než předpokládaný počet spojení na vrcholech v grafu se stejnou distribucí stupňů vrcholů, pak vrcholy komponenty C_c mohou být skutečně součástí validní komunity.

Modularitu komunity c definujeme následně:

$$M_c = \frac{1}{2L} \sum_{(i,j)\in\ C_c} (A_{ij} - p_{ij}), \tag{2.6}$$

kde A_{ij} popisuje skutečné spojení mezi vrcholy i a j, kdežto p_{ij} značí pravděpodobnost existence spojení mezi těmito vrcholy v náhodné síti. Výpočet p_{ij} závisí na volbě nulového modelu, který definuje tvorbu spojení v náhodném grafu. Pro naše potřeby použijeme model Girvanové a Newmana [4], pro který p_{ij} lze zapsat jako:

$$p_{ij} = \frac{k_i k_j}{2L}, \tag{2.7}$$

kde $k_i = \sum_j A_{ij}$ a $L = \sum_{i,j} A_{ij}$.

Pokud je M_c pozitivní, C_c má více hran, než se předpokládalo, může tudíž skutečně tvořit komunitu. Pokud je M_c rovno nule, spojení mezi vrcholy jsou čistě náhodná. Nakonec pro záporné M_c můžeme říct, že vrcholy C_c určitě netvoří komunitu [2].

Pro modularitu můžeme učinit několik pozorování [2]. Pokud bychom pohlíželi na síť jako na jednu velkou komunitu, hodnota M bude rovna nule. Naopak při přiřazení každého vrcholu k vlastní komunitě dostaneme M záporné. Čím vyšší je modularita, tím je rozdělení kvalitnější. Navíc pro menší sítě rozdělení s maximální modularitou pro danou síť odpovídá optimálnímu rozdělení na komunity.

Zmíníme jedno omezení, které má detekce komunit pomocí optimalizace modularity. Maximalizace modularity vede ke slučování malých komunit do větších a nejsme schopní odhalit komunity menší velikosti, než je tzv. rezoluční limit [7]. Skutečné sítě obsahují mnohdy velké množství malých komunit, které budou kvůli rezolučnímu limitu sloučeny do větších, což může vést k určení nesprávné komunitní struktury dané sítě.

Optimalizace modularity je tedy dalším možným způsobem detekce komunit v sítích. V dalším textu si rozebereme Lovaňský algoritmus (Podkapitola 2.3.1), který je založený na tomto přístupu. Kromě detekce komunit je funkce často používána jako měřítko kvality získaných rozdělení na komunity [2].

Na závěr zmíníme, že modularita je specifickým řešením mnohem širšího problému, a to hledání komunit pomocí optimalizace nějaké kvalitativní funkce [2]. Jako příklad si uvedeme algoritmus Infomap [8], který k detekci komunit používá náhodnou procházku grafem. Kvalitativní funkce počítá průměrný počet bitů potřebných k zaznamenání pohybu mezi komunitami a průměrný počet bitů

potřebný k popisu pohybu uvnitř komunit. Pomocí této funkce se snažíme minimalizovat očekávanou délku náhodné procházky v rozdělení na komunity.

2.2.3 Překrývající se komunity

Často vrcholy v síti nejsou součástí pouze jedné komunity. Pokud bychom jako síť vzali všechny děti narozené v roce 2007, potom jedno dítě může patřit hned do několika různých komunit – třída na základní škole, fotbalový tým nebo skupina lidí, která navštěvuje dramatický kroužek. Z tohoto důvodu byly vyvinuty algoritmy na detekci překrývajících se komunit.

My si představíme dva různé přístupy [2]. Prvním bude algoritmus CFinder [9], ve kterém na komunity pohlížíme jako na sjednocení překrývajících se klik. Dvě kliky o velikosti k označíme jako sousední, pokud sdílejí $k-1$ vrcholů. Největší souvislý podgraf získaný sjednocením sousedících *k-klik* nazveme *k-klikovou komunitou*. Typické hodnoty k se pohybují mezi 4 a 6, často je ale třeba k přizpůsobit konkrétním datům.

Druhý algoritmus používá k detekci komunit sdružování hran, což je přístup, který se liší od všech dosud prezentovaných algoritmů. O hranách víme, že mohou v rámci komunitní struktury plnit dvě role – spojovat vrcholy z jedné komunity, nebo spojovat vrcholy z různých komunit. Těchto vlastností se využívá při klastrování pomocí hran (Link clustering algorithm) [10].

Nejprve si definujme podobnost hran. Na její odvození použijeme znalosti o sousedech koncových vrcholů dané hrany a definujeme ji jako poměr velikosti průniku a velikosti sjednocení množin sousedů. V S uvažujeme pouze dvojice hran (i,k) a (j,k) sdílející společný vrchol k.

$$S((i,k),(j,k)) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}, \tag{2.8}$$

kde $n_+(i)$ je množina sousedů vrcholu i zahrnující samotný vrchol i. Takto definované podobnosti se také říká *Jaccardova podobnost* [11], [12].

$S((i,k),(j,k))$ měří relativní počet společných sousedů vrcholů i a j, tudíž $S((i,k),(j,k)) = 1$, pokud mají vrcholy i, j všechny sousedy stejné, a čím menší je $S((i,k),(j,k))$, tím menší je průnik množin sousedů. Tedy $S((i,k),(j,k)) \in [0,1]$.

Zavedení matice S nám umožní použít hierarchické klastrování na nalezení komunit. Postupně budeme tedy spojovat dohromady komunity s nejvyšší podobností hran. Na výstupu dostaneme dendrogram, jehož oříznutím můžeme získat hranové komunity a z nich překrývající se vrcholové komunity.

2.2.4 Komunity v dynamických sítích

Reálné sociální sítě jsou většinou dynamické, tj. neustále se vyvíjí. Přibývají v nich nové vrcholy i nová spojení mezi nimi. Proto se na průzkum komunitní struktury v dynamických sítích používá odlišných metod než pro statické sociální sítě, které byly popsány výše. Existuje několik různých přístupů k detekci komunit v dynamických sítích. Uvedeme několik přístupů detekcí komunit v dynamických sítích, přičemž následujeme práci [13].

Jedním z možných přístupů je dívat se na dynamickou síť jako na statickou a po každé změně hrany nebo vrcholu přepočítat všechny metriky. Takový naivní přístup je ale výpočetně náročný, zejména pro velké sítě. Jako řešení se nabízí procházet síť pouze po určitém (předem daném) intervalu a aplikovat najednou všechny změny, ke kterým za tu dobu v síti došlo. Proto se často omezujeme na tzv. výstřižky, které reprezentují stav sítě v daném časovém intervalu.

První možností je aplikovat na každý výstřižek vybraný algoritmus určený pro statické sítě a následně k sobě přiřadit odpovídající komunity z různých výstřižků. Výhodou je bezesporu možnost použití metod pro detekci komunit a také možnost paralelizace detekce na jednotlivých výstřižcích. Naopak hlavní nevýhodou je, že ne všechny statické algoritmy jsou stabilní. Tím je myšleno, že pro dva různé běhy algoritmu můžeme dostat odlišná rozdělení na komunity. Proto je občas těžké identifikovat, zda rozdíly mezi komunitami dvou různých výstřižků byly skutečně způsobeny změnou ve struktuře sítě nebo jsou pouze pozůstatky různých výsledků statického algoritmu.

Druhý přístup se snaží vyřešit problém s nestabilitou algoritmů a hledáním odpovídajících si komunit z různých výstřižků. Při určování komunit ve výstřižku n použijeme rozdělení na komunity z předchozího výstřižku $n-1$. Tím odpadne nutnost přiřazovat odpovídající si komunity z různých výstřižků. Pro detekci komunit v prvním výstřižku lze navíc použít libovolný efektivní statický algoritmus. Příkladem algoritmu, který využívá tohoto přístupu je DSCAN [14], který podrobněji rozebereme v dalším textu (Podkapitola 2.3.4).

Poslední přístup detekuje komunity procházením všech výstřižků najednou. Existují dva základní způsoby, jak toho lze docílit. Můžeme spojit odpovídající vrcholy z různých výstřižků novými hranami a spustit algoritmus na detekci komunit na této nové velké síti. Možné je také zavést kvalitativní funci a optimalizovat ji na několika vybraných výstřižcích. Kvalitativní funkcí může být například pozměněná modularita. Tento přístup není vhodný pro použití na neustále se měnící sítě, protože při přidání nového výstřižku nelze pouze aktualizovat komunity, ale musí se provést znovu kompletní detekce na všech výstřižcích najednou [13].

2.2.5 Shrnutí

V textu výše je shrnuto několik možných přístupů k detekci komunit. Zmínili jsme dva typy hierarchických algoritmů a jmenovali některé zástupce (G-N [4], Ravazsové [5], Lovaňský [6]). Definovali jsme si kvalitativní funkci zvanou modularita (Vzorec 2.6) a její možné použití na detekci komunit a měření kvality rozdělení na komunity. Představili jsme si dva nejznámnější algoritmy na detekci překrývajících se komunit – CFinder [9] a algoritmus klastrování pomocí hran [10]. Zmínili jsme také základní principy, které se používají při detekci komunit v dynamických sítích.

V další části kapitoly si představíme zástupce každé ze zmíněných kategorií, tj. hierarchický algoritmus, algoritmus založený na optimalizaci modularity, algoritmus pro detekci překrývajících se komunit a algoritmus aplikovatelný na dynamické sítě.

2.3 Vybrané algoritmy

K vlastní implementaci a pozdější aplikaci na vybraná data jsme zvolili následující čtyři algoritmy: Lovaňský algoritmus [6] (hierarchický), algoritmus klastrování pomocí hran [10] (detekce překrývajících se komunit), SCAN (Structural Clustering Algorithm for Networks) [15] a jeho rozšíření pro dynamické sítě DSCAN [14]. Tyto algoritmy zastupují všechny kategorie zmiňované v Podkapitole 2.2.

Jako zástupce hierarchických algoritmů a zároveň algoritmů, které jsou založeny na optimalizaci modularity, jsme zvolili Lovaňský algoritmus. Dle [2] algoritmus je rychlejší, než zbylé zmiňované hierarchické algoritmy (G-N [4], Ravazsové [5]).

Zmiňovali jsme také algoritmus Infomap [8]. S Lovaňským algoritmem [6] mají stejnou složitost $O(Elog(E))$, nebo $O(Nlog(N))$ pro řídké grafy. Jsou porovnatelně rychlé, hierarchické a aplikovatelné na velké grafy. Z některých studií vychází Lovaňský algoritmus z porovnání s algoritmem Infomap jako mírně horší [16], [17], v jiných při měření kvality odlišnými metrikami naopak lepší [18]. V článku [17] je navíc Infomap doporučován primárně k detekci toků informací. V této práci se ale plánujeme zaměřit na porovnávání rozdělení na komunity z hlediska modularity, proto jsme vybrali Lovaňský algoritmus, který je založený na její optimalizaci.

Algoritmus klastrování pomocí hran [10] použijeme na detekci překrývajících se komunit. V části o překrývajících se komunitách jsme zmiňovali i algoritmus CFinder [9], ten má ale kvůli hledání klik vyšší výpočetní složitost než námi vybraný algoritmus.

Algoritmů, které by se zaměřovaly převážně na dynamické sítě je řádově méně než klasických zaměřených na sítě statické. Většinou se detekce komunit provádí aplikací klasických algoritmů na jednotlivé výstřižky. Algoritmus DSCAN [14] jsme zvolili kvůli netradičnímu přístupu a efektivnímu způsobu lokálních aktualizací komunit.

2.3.1 Lovaňský algoritmus

Lovaňský algoritmus [6] je založen na optimalizaci funkce zvané modularita (Vzorec 2.6). V algoritmu ale využijeme její úpravy, a to $\Delta M(C,i)$, která nám udává, o kolik se změní celková modularita přidáním nepřiřazeného vrcholu i ke komunitě C.

$$\begin{aligned}\Delta M(C,i) = &\left[\frac{\sum_{in}+2deg^{C}_{in}(i)}{2m} - \left(\frac{\sum_{tot}+deg(i)}{2m}\right)^2\right] \\ &- \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{deg(i)}{2m}\right)^2\right].\end{aligned} \tag{2.9}$$

$\sum_{in}$ je počet všech hran v rámci komunity C, $\sum_{tot}$ je počet hran incidentních s vrcholy v komunitě C, $deg^{C}_{in}(i)$ je stupeň vrcholu i v rámci komunity C a m je počet všech hran v síti. Pro grafy s váženými hranami počítáme s váženými stupni, tj. stupeň vrcholu je roven součtu vah všech hran incidentních s tímto vrcholem, a v $\sum_{in}$, $\sum_{tot}$ a m za každou hranu přidáme místo 1 její váhu.

Úpravou vzorce 2.9 lze docílit zjednodušené varianty (2.10):

$$\Delta M(C,i) = \frac{2deg_{in}^{C}(i)}{2m} - \frac{\sum_{tot} \cdot deg(i)}{2m^2}. \tag{2.10}$$

Pokud bychom chtěli použít Lovaňský algoritmus na orientované grafy [19], potřebujeme speciálně dodefinovat $\Delta(M_d)$ pro tyto účely (2.11):

$$\Delta M_d(C,i) = \frac{deg_{in}^{C}(i)}{m} - \left[\frac{deg_{out}(i) \cdot \sum_{tot}^{in} + deg_{in}(i) \cdot \sum_{tot}^{out}}{m^2}\right], \tag{2.11}$$

kde $deg_{out}(i)$ a $deg_{in}(i)$ jsou vnější a vnitřní stupeň vrcholu i a $\sum_{tot}^{out}$, $\sum_{tot}^{in}$ počty hran incidentních s vrcholy v komunitě C, které buď vystupují z komunity, nebo do ní vstupují.

Algoritmus probíhá ve dvou krocích, které jsou iterativně opakovány.
1. krok:

- Pro všechny vrcholy grafu lokálně optimalizujeme modularitu;
- Spočítáme dle vzorce 2.10 rozdíl v modularitě, který zle získat přemístěním vybraného vrcholu v do komunity některého z jeho sousedů;
- Najdeme komunitu s maximálním ziskem, od $\Delta(M)$ odpovídajícího této komunitě odečteme snížení, které nastane po odstranění v ze stávající komunity. Pokud $\Delta(M)$ i po tomto odečtení zůstane kladné, přemístíme vrchol v do nové komunity, v ostatních případech ponecháme v ve stávající komunitě;
- Lokální optimalizaci provádíme dokud dochází ke zvyšování M. Jakmile není možné provést tímto způsobem žádné zlepšení, přejdeme na další krok.

2. krok:

- Z komunit získaných v kroku 1 vytvoříme nový graf tak, že vrcholy jsou komunity, hrany jsou sloučením hran mezi komunitami a ke každé komunitě navíc patří smyčka, která má váhu rovnou součtu vah hran v rámci komunity;
- Na tomto nově vytvořeném grafu opět provedeme krok 1, tj. lokální optimalizaci.

Výstupem algoritmu je dendrogram, nové rozdělení do něj přidáváme po každém ukončení prvního kroku algoritmu. Rozdělení s nejvyšší modularitou se nachází vždy na nejvyšší úrovni dendrogramu.

2.3.2 SCAN

SCAN [15] je strukturální algoritmus, který generuje pouze jedno rozdělení na komunity. Od ostatních algoritmů prezentovaných v textu se liší tím, že je schopný identifikovat nejenom jednotlivé komunity, ale i rozcestníky a odlehlé vrcholy. Pro detekci komunit využívá strukturu sítě a propojenosti vrcholů.

Abychom si přiblížili, jak daný algoritmus funguje, je třeba si definovat několik pojmů. Jako $\Gamma(v)$ označme množinu sousedů vrcholu v, do které zahrneme i samotný vrchol v. Nechť $\sigma(v,w)$ je pak strukturální podobnost (*structural similarity*) vrcholů v a w, která je definována následně:

$$\sigma(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{(|\Gamma(v)|)(|\Gamma(w)|)}}. \tag{2.12}$$

Čím více mají vrcholy společných sousedů, tím vyšší je hodnota $\sigma(v,w)$.

ε-okolí vrcholu v je množina jeho sousedů taková, že pro w z této množiny platí: $\sigma(v,w) >= \varepsilon$. Jako jádro označíme vrchol, v jehož ε-okolí leží aspoň μ vrcholů. O vrcholech v, w řekneme, že jsou dosažitelné, pokud existuje posloupnost vrcholů $v_1 = v,v_2,\ldots,v_n = w$, kde pro každou dvojici $v_i - 1$, v_i platí, v_i leží v ε-okolí $v_i - 1$ a navíc $v_1 ... v_n - 1$ jsou jádra. Algoritmus je tedy závislý na dvou vstupních parametrech – ε a μ – pomocí nichž dokážeme regulovat počet nalezených komunit.

Nyní krátce popíšeme princip algoritmu SCAN. Na začátku bereme všechny vrcholy jako nezařazené, tj. není jim přiřazena žádná komunita. Procházíme pak postupně každý nezařazený vrchol a pokud je vrchol v jádrem, založíme novou komunitu. Do této komunity pak ještě přidáme všechny dosud nezařazené vrcholy, které jsou z v dosažitelné. Poté, co jsme prošli všechny vrcholy, mohly nám zbývat nikam nepřiřazené vrcholy, které buď nebyly jádry, nebo neležely v žádném ε-okolí jiného jádra. Takové vrcholy rozdělíme na dva speciální druhy, *rozcestníky* a *odlehlé vrcholy*. Jako rozcestník identifikujeme vrchol, který neleží v žádné komunitě a zároveň sousedí s alespoň dvěma různými komunitami. Odlehlý vrchol taktéž nepatří do žádné komunity, ale sousedí s nejvýše jednou komunitou. Podrobný popis lze nalézt v pseudokódu 1.

Hlavní nevýhoda algoritmu je v nutnosti volby vstupních parametrů uživatelem. Kvůli tomu se musí mnohdy vyzkoušet mnoho možností, než se najde optimální nastavení parametrů pro konkrétní zkoumaná data. Na druhou stranu nám ale umožňuje komplexnější analýzu struktury sítě díky identifikaci rozcestníků a odlehlých vrcholů. Navíc je algoritmus poměrně snadno rozšířitelný na dynamické sítě pomocí algoritmu DSCAN (Podkapitola 2.3.4).

Algorithm 1: SCAN

input: graf $G(V,E)$

```
1  označ všechny vrcholy jako nezařazené;
2  foreach v ∈ V nezařazené do
3      if IsCore (v) then
4          vygeneruj nové IDkomunity;
5          přidej všechny x z ε-okolí v do fronty Q;
6          while Q ≠ ∅ do
7              y = Q.pop;
8              foreach x ∈ DirREACH(y) do
9                  if x je nezařazený nebo neoznačený then
10                     přiřaď IDkomunity k x
11                 if x je nezařazený then
12                     přidej x do fronty Q
13     else
14         označ v jako neoznačený
15 foreach v neoznačený do
16     if v má aspoň dva sousedy patřící do různých komunit then
17         označ v jako rozcestník
18     else
19         označ v jako odlehlý vrchol
```

2.3.3 Algoritmus klastrování pomocí hran

Jak již bylo zmíněno v Podkapitole 2.2.3, algoritmus klastrování pomocí hran [10] využívá k detekci komunit vlastností hran. Algoritmus je hierarchický, výstupem je tedy dendrogram, který na každé úrovni obsahuje jedno rozdělení hran na komunity.

Algoritmus se dá rozdělit na dva kroky.

- **Krok 1:** Spočítáme podobnost pro všechny dvojice hran, využijeme Jaccardovu podobnost (Vzorec 2.8).
- **Krok 2:** Aplikujeme hierarchické klastrování. V každé iteraci vybereme dvojici hran s největší podobností a spojíme komunity, do kterých hrany patří.

Výstupem je podobně jako v případě Lovaňského algoritmu (Podkapitola 2.3.1) dendrogram. Každé rozdělení v dendrogramu ale obsahuje komunity tvořené hranami. Abychom našli rozdělení vrcholů na komunity, je třeba projít všechny vrcholy a přiřadit je ke komunitám, které přísluší hranám incidentním s daným vrcholem.

Jak jsme zmiňovali v Podkapitole 2.2.1, pro získání rozdělení na komunity z dendrogramu je třeba ho na nějakém místě prořezat. Chceme najít především takové rozdělení na komunity, které se nejvíce blíží optimálnímu rozdělení, které

odpovídá skutečné struktuře sítě. Často se k těmto účelům používá modularita, rozdělení na komunity s její nejvyšší hodnotou je označováno za nejlepší [2].

V případě klastrování podle hran ale modularitu použít nelze, protože naše komunity jsou tvořeny hranami a modularitu jsme si definovali na rozdělení vrcholů. Proto autoři v [10] zavedli jiné měřítko pro určení optimálního rozdělení, a to *hustotu rozdělení* D.

Nejprve si definujme *hustotu komunity* c D_c jako počet hran v komunitě c normalizovaný maximálním a minimálním možným počtem vrcholů pro daný počet hran. Nechť m_c je počet hran v komunitě c a n_c počet vrcholů incidentních s těmito hranami, potom

$$\begin{aligned} D_c =& \frac{m_c - (n_c - 1)}{n_c\,(n_c - 1)\,/2 - (n_c - 1)} \\ =& \frac{2\,(m_c - (n_c - 1))}{(n_c - 2)\,(n_c - 1)}. \end{aligned} \tag{2.13}$$

Hustota rozdělení je poté průměrem D_c přes všechny komunity c z rozdělení váženého poměrem m_c vůči počtu všech hran v grafu, M.

$$D = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)\,(n_c - 1)}. \tag{2.14}$$

2.3.4 DSCAN

Poslední z vybraných algoritmů zastupuje kategorii dynamických algoritmů. Jedná se o rozšíření algoritmu SCAN (Podkapitola 2.3.2), od toho také vznikl název algoritmu (Dynamic SCAN) [14].

Algoritmus na vstupu dostává posloupnost výstřižků, kde každý z nich reprezentuje stav sítě v určitém časovém období pomocí grafu. Na první výstřižek aplikujeme algoritmus SCAN (Podkapitola 2.3.2) a najdeme první rozložení vrcholů na komunity.

Každý další výstřižek porovnáme s předchozím a následně na každou nově přidanou (nebo naopak odebranou) hranu aplikujeme aktualizační funkci. Tato funkce aktualizuje stav vrcholů, které byly zasaženy změnou hrany, konkrétně upraví komunitu, do které každý z vrcholů patří. Pokud zjistíme, že vrchol v novém grafu nepatří do žádné komunity, prohlásíme ho za odlehlý vrchol, nebo rozcestník. Algoritmus tedy provádí změny pouze lokálně, což umožňuje jednodušší sledování vývoje komunit v čase.

Na výstupu pro každý výstřižek zvlášť dostaneme příslušné rozdělení vrcholů na komunity.

3. Analyzovaná data

Jako data k podrobnější analýze a aplikaci vybraných algoritmů na detekci komunit jsme zvolili dataset e-mailů zaměstnanců firmy *Enron* [20]. Níže uvedené informace o firmě jsme čerpali z článků [21] a [22].

Firma Enron byla založena v roce 1985 Kennethem Layem sloučením dvou společností na distribuci zemního plynu *Houston Natural Gas* a *Internorth*. Díky inovacím Jeffreye Skillinga se firma Enron dostala do vedoucí pozice ve svém odvětví. Působení firmy se rozšířilo z distribuce zemního plynu a obchodování s různými druhy energií i na sféru služeb. Jedním z nejlepších obchodníků firmy byl Andrew Fastow, který se nakonec vypracoval na ředitele financí a měl na starost financování společnosti a investování do stále složitějších produktů.

V průběhu 90. let se vedení společnosti dopustilo několika špatných podnikatelských rozhodnutí, která vedla k zadlužení společnosti. Tento fakt se snažilo vedení firmy zamaskovat před investory, a to především podvodnými účetními praktikami, kterým se také často říká „kreativní účetnictví“. Tyto praktiky umožňovaly firmě připisovat si do aktuálních příjmů budoucí příjmy z obchodů, a tím získat iluzi vyšších současných příjmů. Krycí peněžní operace byly prováděny s SPE (*special purpose entities*), společnostmi zakládanými pouze za účelem transferu peněz. Mnoho z vytvořených SPE bylo psáno na Andrewa Fastowa.

Vážnost situace začala být zřejmá v polovině roku 2001, když někteří analytici začali zkoumat detaily ve veřejně dostupných finančních operacích firmy Enron. Krátce poté zahájila Komise pro cenné papíry (*Securities and Exchange Commission*, SEC) interní vyšetřování pro podezření z podvodů, během kterého byly odhaleny operace mezi firmou Enron a Fastowem. Po zveřejnění detailů účetních podvodů klesla tržní cena akcií firmy z 90 dolarů na méně než jeden. Začátkem prosince 2001 požádala firma Enron o ochranu před věřiteli a následně oznámila bankrot.

Mnoho vedoucích představitelů firmy bylo později za podvody obžalováno a odsouzeno. Většina akcionářů podala žaloby na společnost Enron a na firmu *Anderson Consulting*, která v Enronu prováděla audit. Kauza způsobila v USA změnu zákonů na vykazování finanční činnosti pro veřejně obchodující firmy, zároveň vedla k zavedení tvrdých trestů za jakékoliv podvody v účetnictví.

3.1 Dataset Enron

Data, která tvoří zkoumaný dataset, byla původně zveřejněna federální komisí pro regulaci energií (*Federal Energy Regulatory Commission*), následně byla zpracována CALO Projektem (*Cognitive Assistant that Learns and Organizes*), který poskytl dataset zpracovaný do současné podoby. Dataset obsahuje okolo půl milionu e-mailových zpráv, které jsou roztříděné do 150 složek. Každá složka nese jméno zaměstnance, zřejmě z vyššího managementu firmy. V datasetu je zachyceno období před zánikem firmy (od 30. října 1998 do 12. července 2002).

Každá e-mailová zpráva v datasetu má přibližně stejný tvar hlavičky, který obsahuje následující položky popsané na Obrázku 3.1. Pro nás jsou zajímavé především položky `Datum` (Date), `Odesílatel` (From), `Příjemce` (To), `Kopie` (Cc) a `Předmět` (Subject). Příklad e-mailové hlavičky lze najít na Obrázku 3.3.

```
Message-ID: ID
Date: Den_v_Týdnu, Den Měsíc Rok Hod:Min:Sek -0800 (PST)
From: e-mail odesilatele
To: e-maily příjemců
Subject: předmět zprávy
CC: e-maily příjemců v kopii
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

Obrázek 3.1: Popis e-mailové hlavičky e-mailů z datasetu Enron.

3.2 Předzpracování dat z datasetu Enron

3.2.1 Zpracování dat

Vytvořili jsme rozsáhlou sociální síť, která obsahovala všechny uživatele, kteří se vyskytli v hlavičce kteréhokoliv e-mailu. Z každého e-mailu jsme získali informace o odesílateli, příjemci a také zda uživatel dostal zprávu v kopii, nebo jako přímý adresát.

Při zpracování každé ze 150 složek datasetu jsme se snažili maximálně omezit příjemce a adresáty na zaměstnance firmy Enron, tj. zajímaly nás především adresy z domény *@enron*.

Jako identifikátory vrcholů v síti si udržujeme jednotlivé e-mailové adresy. Původně jsme uvažovali i nad přiřazením jednotlivých adres ke jménům uživatelů, ale nejenom, že jejich zápis v jednotlivých souborech neměl jednotný formát, ale v mnoha zprávách nebyla jména k adresám přiřazena vůbec . Proto jsme se rozhodli pro extrahování pouze e-mailových adres.

V každém souboru jsme našli řádky s odpovídajícím začátkem (klíčová slova Datum, Odesílatel, Příjemce, Kopie, Předmět) a zpracovali jejich obsah. Každou novou adresu jsme uložili do množiny unikátních adres. Na ukládání hran jsme použili `slovník`, kde jako klíče nám slouží dvojice `(odesílatel, příjemce)` a jako hodnoty unikátní množiny trojic `(datum, čas, je/není v kopii)`. Příklad záznamu ve slovníku lze vidět na Obrázku 3.2.

```
(skilling, dasovich) : (14.02.2000, 15:02:45, False)
```

Obrázek 3.2: Příklad záznamu v slovníku použitém na uložení informací o posílaných zprávách mezi uživateli.

Z výše popsaných struktur jsme následně vytvořili následující grafy. Prvním je statický neorientovaný graf s váženými hranami. Váhy byly spočítany jako vážený součet všech hran mezi danými dvěma vrcholy v původní síti, která má podobu multigrafu. Za každou hranu reprezentující přímou zprávu jsme započítali do celkové váhy 2, za zprávu v kopii váhu 1. Druhým je statický orientovaný graf s váženými hranami. Váha každé hrany byla vypočítána stejně jako u prvního grafu.

```
Message-ID: <7618763.1075855377753.JavaMail.evans@thyme>
Date: Mon, 31 Dec 2001 10:53:43 -0800 (PST)
From: louise.kitchen@enron.com
To: wes.colwell@enron.com, georgeanne.hodges@enron.com, rob.milnthorp@enron.com,
     john.zufferli@enron.com, peggy.hedstrom@enron.com,
     thomas.myers@enron.com, s..bradford@enron.com, lloyd.will@enron.com,
     sally.beck@enron.com, m.hall@enron.com, m..presto@enron.com,
     david.forster@enron.com, leslie.reeves@enron.com,
     chris.gaskill@enron.com, robert.superty@enron.com,
     fred.lagrasta@enron.com, laura.luce@enron.com,
     barry.tycholiz@enron.com, brian.redmond@enron.com,
     frank.vickers@enron.com, c..gossett@enron.com, john.arnold@enron.com,
     mike.grigsby@enron.com, k..allen@enron.com, scott.neal@enron.com,
     a..martin@enron.com, s..shively@enron.com, rita.wynne@enron.com,
     jenny.rub@enron.com, jay.webb@enron.com, e..haedicke@enron.com,
     rick.buy@enron.com, f..calger@enron.com, david.duran@enron.com,
     mitch.robinson@enron.com, mike.curry@enron.com,
     tim.heizenrader@enron.com, tim.belden@enron.com, w..white@enron.com,
     d..steffes@enron.com, c..aucoin@enron.com, a..roberts@enron.com,
     david.oxley@enron.com
Subject: NETCO
Cc: john.lavorato@enron.com
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

Obrázek 3.3: Ukázka e-mailové hlavičky z jednoho z e-mailů z datasetu Enron (`allen-p/inbox/13`).

3.2.2 Analýza vytvořených grafů

Grafy zmíněné na konci předchozí sekce jsme zanalyzovali blíže, abychom odstranili případné duplicitní vrcholy, nebo například vrcholy s příliš malým stupněm na okraji sítě, které na celou strukturu nebudou mít téměř žádný vliv. Adresy reprezentující vrcholy vytvořených grafů jsme proto rozdělili na dva druhy – adresy patřící zaměstnancům z vyššího managementu firmy a ostatní uživatele. Zaměstnanci vyššího managementu tvoří pomyslné jádro celé naší sítě, proto je zařadíme do výsledného grafu.

Jak již bylo řečeno výše, dataset se skládá ze 150 složek, kde každá patří některému z bývalých zaměstnanců vyššího managementu Enronu. Tuto skupinu zaměstnanců jsme stanovili jako množinu základních uživatelů e-mailové sítě firmy Enron. Každému uživateli byla přiřazena množina e-mailových adres, ve kterých se vyskytovala nějaká kombinace jejich jména a příjmení. Při zpracování jsme odhalili, že některé složky obsahují stejné uživatele, proto konečný seznam obsahoval pouze 148 jmen. Seznam uživatelů z množiny základních uživatelů i s příslušnými adresami lze nalézt v souboru `main_users.txt`.

Pro adresy mimo množinu základních uživatelů jsme vytvořili tři kategorie: *odesílatelé*, *příjemci* a *příjemci v kopii*, přičemž každá adresa patří alespoň do jedné z těchto kategorií. Spočítali jsme výskyty jednotlivých adres v e-mailech z celého datasetu pro každou kategorii. Většina adres se v datasetu vyskytuje jen zřídka, pro více než 90 % adres platí, že z nich bylo odesláno, nebo na ně přijato méně než 50 zpráv. Průměrný počet odeslaných zpráv je 14, přijatých 38 a v kopii 20. Tyto i další statistiky získané z tohoto datasetu jsou v Tabulce 3.1.

Kategorie	MIN	MAX	Průměr	Modus	Medián
Odesílatelé	1	9149	14,52	1 [7168]*	3
Příjemci	1	9281	38,34	1 [15384]*	3
Příjemci v kopii	1	9093	20,07	1 [5180]*	3
Všichni	1	27523	39,09	1 [21871]*	3

* Počet výskytů prvku.

Poznámka: Mezi adresami nebyly započítávány ty, které patří uživatelům z množiny základních uživatelů.

Tabulka 3.1: Statistiky odeslaných a přijatých zpráv na e-mailovou adresu v rámci kategorií.

Pozorování z předchozího odstavce nás vedla k vytvoření menších sociálních sítí, ve kterých jsme se rozhodli ponechat pouze množinu základních uživatelů a uživatele mimo tuto množinu, kteří s jinými uživateli sítě komunikovali nejvíce.

3.3 Vytvoření menších sociálních sítí

3.3.1 Grafy na množině základních uživatelů

Množinu vrcholů pro první dva grafy tvoří množina základních uživatelů. Mají 148 vrcholů, 2094 hran pro orientovaný graf a 1535 pro neorientovaný. Hrany jsou vážené a váhy byly spočítány jako vážený součet všech hran mezi danými dvěma vrcholy v původní síti, která má podobu multigrafu. Za každou hranu reprezentující přímou zprávu jsme započítali do celkové váhy 2, za zprávu v kopii váhu 1.

Orientovaný graf

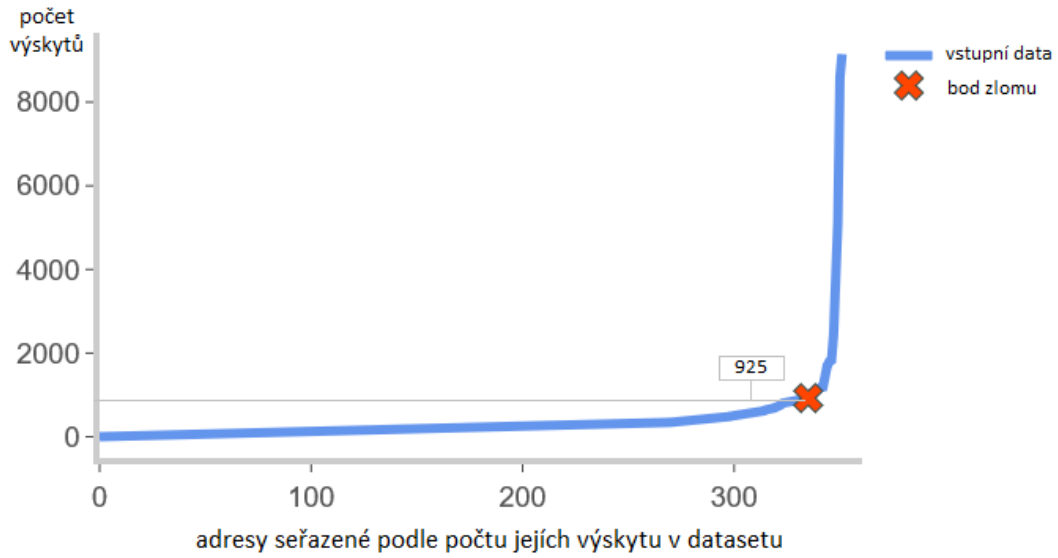
V orientovaném grafu patří vrchol s největším váženým výstupním stupněm (7288) uživateli *dasovich*. Nejvyšší vstupní (4770) i celkový stupeň (9826) má vrchol uživatele *kaminski*. Střední vážený vstupní nebo výstupní stupeň vrcholu je 367,479, střední celkový stupeň pak 734,959.

Orientovaný graf

V neorientovaném grafu je vrchol s nevětším váženým stupněm (9826) stejný jako v případě orientovaného grafu, tj. patří uživateli *kaminski*. Střední stupeň vrcholu je 20,743, střední vážený stupeň vrcholu potom 734,878.

3.3.2 Rozšíření množiny základních uživatelů

Kromě grafů vytvořených z množiny základních uživatelů jsme se rozhodli pro vytvoření o něco větších grafů rozšířením základní množiny o další uživatele. Jak jsme viděli v Tabulce 3.1, většina adres se v datasetu vyskytuje jen pákrát, proto vybereme pouze adresy s nejčastějším výskytem.



Obrázek 3.4: Počty výskytů adres z množiny odesílatelů i s nalezeným zlomovým bodem (925).

Pro oříznutí jsme použili tzv. *pravidlo lokte* (*knee method*, nebo také *elbow rule*) [23]. Jednoduše řečeno je to metoda, která dokáže najít tzv. bod zlomu funkce. Pro určení bodu zlomu jsme použili online aplikaci `ikneed` od *Kevina Arvaie* [24].

Body zlomu jsme hledali zvlášť pro každou kategorii zmíněnou v Tabulce 3.1. Jako data reprezentující hodnoty funkce jsme vzali setříděnou množinu počtů výskytů jednotlivých adres. Na Obrázku 3.4 jsou znázorněna data se zlomovým bodem pro kategorii odesílatelů, kde bod zlomu je při počtu výskytů 925. Počet výskytů ve zlomových bodech pro všechny kategorie lze nalézt v Tabulce 3.2.

Odesílatelé	Příjemce	Příjemce v kopii
925	2306	1098

Tabulka 3.2: Zlomové body pro jednotlivé kategorie.

Do množiny adres, které budou patřit uživatelům z rozšíření, jsme tedy přidali z každé kategorie pouze ty, které měly počet výskytů vyšší než příslušný bod zlomu.

Z výsledných grafů jsme ještě několik vrcholů odstranili, protože byly buď plně izolované (uživatel nejspíš nekomunikoval s nikým z množiny základních uživatelů), nebo měly příliš nízký stupeň a sloužily pouze jako stoky. Jednalo se o následující adresy: *no.address@enron.com, enron.announcements@enron.com, announcements.enron@enron.com*. Jak můžeme vidět, jsou to adresy nepatřící žádnému konkrétnímu uživateli. Pro větší přehlednost a jednodušší reprezentaci uživatelů ve výsledných grafech jsme vytvořili pro každou adresu z předpokládaného příjmení uživatele unikátní token.

Výsledné rozšířené grafy jsou tedy tvořeny 245 vrcholy, orientovaný 3626 hranami a neorientovaný 2978 hranami. Po analýze výsledných grafů jsme navíc odstranili následující adresy, protože neinteragovaly s žádnou z rozšířené

množiny uživatelů. Jednalo se o následující adresy: *all.worldwide@enron.com, jbryson@enron.com, dporter3@enron.com, recipients@enron.com*. Výše uvedené počty vrcholů a hran platí pro grafy s již odstraněnými vyjmenovanými uživateli. Hrany v grafech jsou podobně jako v grafech ze základní množiny uživatelů vážené.

Orientovaný graf

Na předních příčkách, podobně jako u grafu ze základní množiny uživatelů, jsou uživatel *dasovich*, kterému patří vrchol s nevětším váženým výstupním (28504) a celkovým stupněm (29579), a uživatel *kaminski* s největším vstupním stupněm (4807). Střední vážený vstupní stupeň vrcholu je stejný jako výstupní, a to 452,016, střední celkový stupeň pak 904,032.

Neorientovaný graf

Vrchol s největším váženým stupněm (29579) je jako i v případě orientovaného grafu uživatel *dasovich*. Střední stupeň vrcholu je 24.385, střední vážený stupeň vrcholu je 903.983.

3.4 Dynamický graf

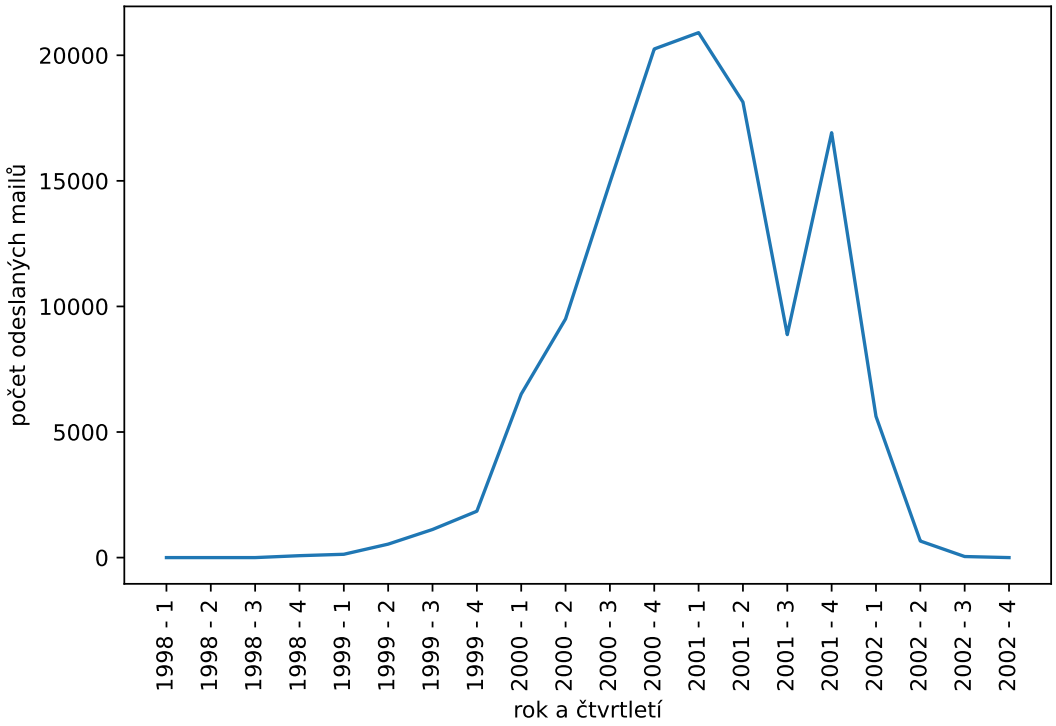
Jak již bylo popsáno výše, dataset, se kterým pracujeme, byl vytvořen po zániku firmy Enron a obsahuje e-mailové zprávy z posledních několika let před zánikem. Proto prozkoumáme nejenom statické komunity, ale i jejich vývoj v čase a zkusíme popsat, jaký vliv na strukturu komunit měly události, které firmu zasáhly.

Dataset obahuje zprávy z období od 30. října 1998 do 12. července 2002. Dynamický graf budeme reprezentovat pomocí výstřižků zmíněných v Podkapitole 2.2.4. Protože výše zmíněné období není příliš dlouhé a, jak je vidět na grafu z Obrázku 3.5, převážná většina zpráv je koncentrována do konce sledovaného období, rozhodli jsme se pro rozdělení zpráv z datasetu dle jednotlivých čtvrtletí. Celkem máme tedy vytvořenou posloupnost 15 grafů.

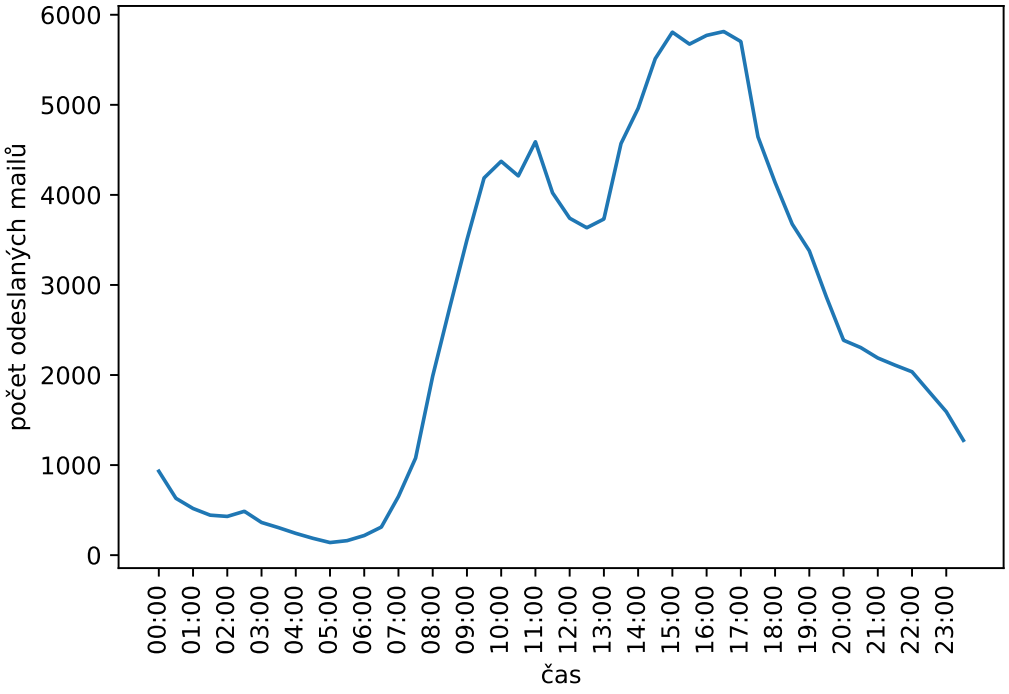
3.5 Grafy reprezentující stav sítě v různou denní dobu

Kromě vývoje sítě v čase jsme se rozhodli zanalyzovat, jak se mění komunity v průběhu různých denních dob. Pro tyto účely jsme den rozdělili na několik částí. Rozdělení je založeno na rozložení posílaných zpráv v jednotlivých časových obdobích, které je vidět na grafu z Obrázku 3.6.

Na grafu jsou patrná dvě významná lokální maxima mezi 9. a 11. hodinou a mezi 14. a 17. hodinou. Tato maxima odpovídají průměrnému dnu zaměstnance ve velké firmě – období práce do oběda a po obědě, kde jako období oběda identifikujeme propad mezi těmito dvěma maximy (lokální minimum okolo 12:30).



Obrázek 3.5: Rozložení poslaných mailů dle čtvrtletí od roku 1998 do konce roku 2002.



Obrázek 3.6: Rozložení poslaných mailů v rámci dne. Hodnoty jsou souhrnem pro celý dataset.

Proto jsme se rozhodli rozdělit den na 3 části. Období od 6:00 do 12:30 si pojmenujme jako *ranní pracovní dobu*, od 12:30 do 20:00 jako *odpolední pracovní dobu* a zbytek z rozmezí 20:00 až 6:00 jako *noční zprávy*.

Při uvažování nad běžným dnem zaměstnance pracujeme s předpoklady, že ráno zaměstnanec při příchodu do práce posílá především pracovní e-maily, během poledne se domlouvá s kolegy na obědu a odpoledne se opět věnuje pracovním e-mailům, nebo s kolegy probírá plány na večer a případné jiné záležitosti. Proto předpokládáme, že se skupiny lidí, se kterými se zaměstnanec baví, mohou v průběhu dne měnit.

3.6 Shrnutí

Data, na kterých budeme podrobněji analyzovat detekci komunit algoritmy představenými v druhé kapitole (Podkapitola 2.3), pocházejí z datasetu e-mailových zpráv mezi zaměstnanci firmy Enron [20]. Společnost se zabývala prodejem a distribucí různých druhů energií. Po několika neúspěšných investicích se firma zadlužila a vedení společnosti za účelem skrytí klesajících příjmů před věřiteli a investory se uchýlilo k podvodným praktikám. Veškeré podvody nakonec byly odhaleny a došlo k zániku firmy. Část e-mailové komunikace mezi zaměstnanci z posledních let existence firmy byla později zveřejněná k vědeckým účelům.

Z datasetu jsme extrahovali celkem 4 menší grafy na dvou množinách uživatelů (Podkapitola 3.3). Množinu základních uživatelů tvoří 148 zaměstnanců a na této množině jsme vytvořili orientovaný a neorientovaný graf komunikace mezi jednotlivými zaměstnanci. Základní množinu uživatelů jsme následně rozšířili o další uživatele, o kterých jsme zjistili, že posílali nejvíce zpráv. Na této rozšířené množině jsme vytvořili opět jeden orientovaný a jeden neorientovaný graf. Na těchto grafech budeme detekovat komunity pomocí tří z popsaných algoritmů (Lovaňský – Podkapitola 2.3.1, SCAN – Podkapitola 2.3.2 a algoritmus klastrování pomocí hran – Podkapitola 2.3.3).

Kromě statických grafů jsme vytvořili i jeden dynamický (Podkapitola 3.4) na rozšířené množině uživatelů. Na něm plánujeme s pomocí algoritmu DSCAN (Sekce 2.3.4) sledovat vývoj struktury komunit v síti v čase.

Kromě zmíněných grafů máme připravené i grafy, které reprezentují stav sítě v různou denní dobu (Podkapitola 3.5). Tyto grafy později použijeme především k určení reprezentativnosti komunit detekovaných vybranými algoritmy.

4. Experimenty

V předcházejících kapitolách jsme si představili algoritmy na detekci komunit v sociálních sítích (Podkapitola 2.3) a data (Kapitola 3), ze kterých jsme vytvořili několik grafů reprezentujících sociální síť. V této kapitole se budeme zabývat aplikací představených algoritmů na připravená data. Postupně představíme výsledky pro jednotlivé algoritmy, zhodnotíme, jak si na vybraných grafech vedly a na závěr provedeme jejich vzájemné porovnání.

4.1 Sledované vlastnosti detekovaných komunit

Každý z algoritmů představených v Podkapitole 2.3 funguje na jiných základech a je potřeba si stanovit parametry, které pak poslouží k porovnání výsledných rozložení na komunity získaných těmito algoritmy. Sledované parametry jsme rozdělili pro přehlednost na několik kategorií – základní parametry, struktura komunit a charakter komunit. Souhrn všech sledovaných parametrů lze nalézt v Tabulce 4.1.

Mezi *základní parametry* jsme zařadili počet komunit v rozdělení a velikost komunit, která odpovídá počtu vrcholů v dané komunitě. Kromě toho je zajímavé zjistit i průměrnou velikost komunit a rozptyl těchto velikostí. Dále jsme mezi základní parametry zařadili průměr komunity, který si definujeme jako nejdelší z nejkratších vzdáleností mezi libovolnými dvěma vrcholy komunity, a hustotu komunity, kterou budeme měřit jako počet hran v rámci komunity normalizovaný maximálním možným počtem hran na množině vrcholů dané komunity. Tyto parametry nám pomohou určit, zda má algoritmus tendenci rozdělovat graf na komunity přibližně stejné velikosti či nikoliv.

Další kategorie, kterou jsme označili jako *strukturu komunit*, se zabývá charakterem vrcholů v komunitě, a to především proměnlivosti a zastoupením důležitých vrcholů v ní. Důležitost budeme měřit pomocí tří centralit představených v Podkapitole 2.1.1 – D-centrality, C-centrality a B-centrality.

D-centralitu využijeme na odhalení struktury komunity, zda obsahuje relativně vyvážené členy, nebo naopak pár dominantních a zbytek pouze navázaný na ně. C-centralita nám ukáže vrcholy, které můžeme označit jako centra komunit, protože vysoká C-centralita odpovídá krátké vzdálenosti k ostatním vrcholům v komunitě. A nakonec pomocí B-centrality odhalíme vrcholy, jejichž odstranění může vést například i k rozpadu komunity, protože vrcholy s vysokou B-centralitou jsou ty, kterými prochází nejvíce spojení v rámci komunity.

Další kategorie, *charakter komunit*, se zabývá kvalitou nalezených komunit. Pro měření využijeme hlavně modularitu celého rozdělení vrcholů na komunity. Zde je třeba podotknout, že výsledky algoritmu SCAN (Podkapitola 2.3.2) budou mít obecně nižší modularitu v porovnání třeba s Lovaňským algoritmem z toho důvodu, že SCAN pracuje i s odlehlými vrcholy a rozcestníky, které nepatří do žádné komunity. Kvůli tomu bude velikost komunit nalezených algoritmem SCAN obecně menší, a proto bude menší i celková modularita.

Rovněž zkusíme odhadnout, zda jsou komunity odpovídající (reprezentativní), tj. mají nějaký základ ve skutečném světě. Konkrétně budeme hledat společná témata posílaných zpráv analýzou předmětů zpráv. Pokud si například uživatelé

často píší o Kalifornii a energii, nejspíš odpovídá komunita oddělení firmy, které se zabývá prodejem a distribucí energií v rámci Kalifornie. Nebo zkusíme najít souvislost v grafech denních dob (Podkapitola 3.5), což by napovídalo tomu, že je komunita skupinou lidí, kteří se setkávají například na obědech, tj. mimo pracovní dobu jsou přátelé.

základní parametry	struktura komunit	charakter komunit
počet komunit	D-centralita	modularita
průměrná velikost komunit	C-centralita	reprezentativnost
rozptyl velikostí	B-centralita	
průměr komunity		
hustota komunity		

Tabulka 4.1: Přehled parametrů sledovaných ve výsledných rozděleních grafů na komunity rozdělený dle zavedených kategorií.

4.2 Experimenty s Lovaňským algoritmem

V předchozím textu popsaný Lovaňský algoritmus (Podkapitola 2.3.1) je založen na iterativní optimalizaci modularity (Vzorec 2.6). Výstupem je dendrogram rozdělení na komunity, kde nejvyšší úroveň dendrogramu obsahuje rozdělení s nejvyšší modularitou.

V algoritmu lze nastavit několik parametrů (Příloha A.2.1). Prvním je rezoluční parametr — čím je vyšší, tím méně jsou malé komunity slučovány do větších, tedy jinými slovy algoritmus najde více menších komunit. Posouváním výšky modularity bychom tedy chtěli předejít sloučení většiny vrcholů do jedné velké komunity a najít nějakou optimální hodnotu.

Pomocí dalšího parametru lze nastavit randomizované procházení vrcholů v prvním kroku algoritmu. Výsledky pokusů pro různá semínka budou prezentovány po zprůměrování.

Dále lze algoritmus také modifikovat tak, aby při výpočtech bral ohled na váhy hran. Předpokládáme, že započítání vah povede ke zvětšení počtu komunit a ke zmenšení jejich velikostí.

Pro porovnání výsledků běhů algoritmu s různými parametry použijeme celkovou modularitu rozdělení vrcholů na komunity, počet detekovaných komunit, průměr a rozptyl jejich velikostí, dále také hustotu a průměr detekovaných komunit. Pro určení reprezentativnosti komunit použijeme grafy zobrazující komunikaci uživatelů v různou denní dobu a předměty posílaných zpráv mezi uživateli.

4.2.1 Výsledky získané na neorientovaném grafu základních uživatelů

Graf základních uživatelů, jak již bylo uvedeno v Podkapitole 3.3.1, tvoří 148 vrcholů a 1535 hran. Každý vrchol v průměru sousedí s 21 jinými vrcholy a průměrný vážený stupeň se blíží 734,88.

Dendrogram rozdělení na komunity reprezentuje hierarchickou strukturu komunit v grafu, ale během experimentů jsme zjistili, že graf základních uživatelů zřejmě žádnou významnou hierarchickou strukturu neobsahuje. Počet úrovní dendrogramu se pohyboval mezi 2 a 3, přičemž počet komunit dvou nejvyšších úrovní se obvykle lišil velmi málo, o jednu nebo dvě komunity. Proto pro porovnání různých nastavení nebudeme hierarchickou strukturu brát v úvahu a počítat budeme pouze s rozdělením vrcholů na komunity s nejvyšší modularitou.

Rozdělení vrcholů na komunity bez použití randomizace

Nejprve si představíme techniky, kde nebyla použita randomizace. Budeme sledovat především změny způsobené zvyšováním rezolučního parametru a to, jakým způsobem tyto změny ovlivní započítávání vah. Jako základní rozdělení uvažujme výsledek s rezolucí nastavenou na 1 a bez započítávání vah. Výsledky jednotlivých pokusů lze nalézt v Tabulce 4.2, kde jsou navíc zvýrazněny nejlepší hodnoty modularity, rozptylu velikostí komunit, průměr a hustota komunit.

R	M	Mw	NC	NCw	S	Sw
1	0.03	0.64	4	31	37	4.77
2	**0.31**	0.65	15	28	9.87	5.29
5	0.25	0.65	19	23	7.79	6.43
10	0.27	**0.66**	19	22	7.79	6.72
20	0.27	0.66	19	22	7.79	6.72

R	V	Vw	Diam	Diamw	Dens	Densw
1	3408.5	**36.88**	1.5	**1.00**	0.74	**0.83**
2	147.32	56.13	1.53	1.04	0.72	0.81
5	**116.69**	82.16	1.37	1.22	0.76	0.78
10	123.85	91.93	**1.32**	1.23	**0.78**	0.77
20	123.85	95.65	1.32	1.18	0.78	0.77

Poznámka: tučně zvýrazněny jsou nejlepší hodnoty.
R–použitý rezoluční parametr, M–modularita,
NC–počet komunit, S–průměrná velikost komunity,
V–rozptyl velikosti komunit, w–započítáváme váhy hran,
Diam–průměrný průměr komunity, Dens–průměrná hustota komunity.

Tabulka 4.2: Rozdělení získaná Lovaňským algoritmem na neorientovaném grafu základních uživatelů.

Jak lze vidět, váha hran výrazně ovlivňuje kvalitu i charakter rozdělení na komunity. Předpokládáme, že je to způsobené především tím, že v e-mailové síti uživatelů hraje roli spíš kvalita spojení, než jeho existence. V našem případě je kvalita vyjádřena váhou. Rozdělení na komunity, kde jsme započítávali váhy, mají vyšší modularitu (0,66) a vyšší počet menších komunit bez výraznějších výkyvů ve velikosti (počet komunit se napříč experimenty pohybuje mezi 22 a 31 a průměrná velikost komunity pro všechny experimenty je v rozmezí 4-7 vrcholů). Tím je také způsobená vyšší hustota (v nejlepším případě nad 80 %) a menší průměrné průměry komunit, které nepřevýšily hodnotu 1,25.

Pro neohodnocený graf se podařilo dosáhnout nejlepší modularity s hodnotou 0,31. Počet komunit se pohyboval okolo 17 a průměrná velikost komunity okolo 8, ale zaznamenali jsme mnohem větší výkyvy oproti ohodnocenému grafu. Vysoký rozptyl velikostí je způsoben především tím, že mnoho vrcholů je přiřazeno do jedné velké centrální komunity. Tato vlastnost je ale pro sociální sítě běžná a protože taková dominantní komunita vznikla pro veškeré možné hodnoty rezolučního parametru, můžeme předpokládat, že i v síti uživatelů firmy Enron platí tento jev. Kvůli větší velikosti komunit se zvyšoval i jejich průměr, který se pohyboval mezi 1,32 a 1,5, a nižší byla i hustota (okolo 75 %).

Změny rezolučního parametru jsou nejlépe viditelné na výsledcích neohodnoceného grafu, nejvíce pro hodnoty *res1* a *res2*. Základní rozdělení sloučilo přes 93 % všech vrcholů, konkrétně 138, do jedné jediné komunity. Zvýšení rezolučního parametru pomohlo v rozdělení této obrovské komunity na menší, které lépe reflektují skutečnou strukturu sítě, navíc nám řádově vzrostla modularita rozdělení. V ohodnoceném grafu změna rezoluce měla opačný efekt, a to snížení počtu komunit. Změny ale nebyly tak markantní jako v případě neohodnoceného grafu.

Reprezentativnost rozdělení vrcholů na komunity

Při podrobnější analýze rozdělení na ohodnoceném a neohodnoceném grafu vyšlo najevo, že se základy větších komunit od sebe příliš neliší (porovnání bylo provedeno pro výsledky s rezolučním parametrem 5, protože u nich se k sobě poprvé výrazněji blíží počty komunit v obou rozděleních). U ohodnocených grafů dochází k tomu, že některé vrcholy nemají zřejmě příliš silná spojení s žádnou větší komunitou, a proto jsou ponechány izolovaně. Z těchto důvodů budeme zkoumat reprezentativnost pouze na jednom ze zmíněných grafů, konkrétně na ohodnoceném, protože lépe reflektuje skutečnou podobu sítě.

Pro tyto účely jsme vybrali subjektivně nejlepší rozdělení vrcholů na komunity. Chceme vybrat rozdělení s co nejlepšími parametry a zároveň takové, aby se strukturou (především počtem komunit) blížilo rozdělení na neohodnoceném grafu. Kvůli poslednímu kritériu nám nevyhovuje základní rozdělení, i když má nejlepší parametry. Pro další analýzu jsme nakonec zvolili rozdělení s menším počtem komunit, s větší průměrnou velikostí a stále poměrně vysokou hustotou, tedy rozdělení s rezolucí nastavenou na 2.

Pro analýzu jsme vybrali tři komunity o velikostech 19, 11 a 10. Označme je popořadě A, B, C, na Obrázku 4.1 jsou značeny zelenou, modrou a oranžovou barvou. Průměr každé z nich je 2 a hustoty mají následující hodnoty: A–37 %, B–67 %, C–75 %. Reprezentativnost budeme zkoumat především na jádrech komunit, která sestavíme z nejdůležitějších vrcholů v síti. Důležitost měříme pomocí B–, C– a D–centrality. Tabulka 4.3 obsahuje uživatele s nejvyššími hodnotami centralit pro jednotlivé komunity. K analýze reprezentativnosti použijeme tři grafy komunikace mezi uživateli z různých denních dob a předměty posílaných zpráv.

Z grafů denních dob (Podkapitola 3.5) se, bohužel, nepodařilo vyčíst žádný výrazný vzor, který by danou komunitu definoval. Data z nočních hodin ale potvrzují nalezenou strukturu komunit tím, že v této denní době probíhala komunikace, až na několik málo slabších spojení, výhradně uvnitř komunity.

Při analýze předmětů zpráv jsme zjistili, že mnoho zpráv neobsahovalo žádný předmět. Přesto se podařilo existenci komunit v reálném životě podložit. Při analýze jsme postupovali následujícím způsobem: jako první jsme vytvořili seznamy

Centralita	Uživatelé z komunity A
D–centralita	williams_w3, symes, guzman, slinger, solberg
B–centralita	williams_w3, symes, guzman, slinger, salisbury
C–centralita	williams_w3, symes, guzman, slinger, salisbury
	Uživatelé z komunity B
D–centralita	allen, grigsby, kuykendall, tholt, reitmeyer
B–centralita	allen, grigsby, gay, kuykendall, tholt
C–centralita	allen, grigsby, kuykendall, tholt, reitmeyer
	Uživatelé z komunity C
D–centralita	jones, taylor, shackleton, heard, clair
B–centralita	jones, taylor, clair, sager, shackleton
C–centralita	jones, taylor, clair, shackleton, heard

Tabulka 4.3: Vrcholy s nejvyšší hodnotou centralit pro vybrané komunity.

Klíčové slovo	Množiny komunity A
meeting	williams_w3, slinger, symes
baseball game	williams_w3, symes, slinger
empower	williams_w3, symes, solberg, slinger, linder
	Množiny komunity B
meeting	kyukendall, tholt, allen, grigsby, south, ermis holst, reitmeyer, gay, smith
gas	tholt, allen, grigsby, south, ermis, holst reitmeyer, gay, smith
	Množiny komunity C
meeting	jones, panus, schackleton, taylor, clair, sager dickson, haedicke, bailey
lunch	sager, schackleton, clair, taylor, jones

Tabulka 4.4: Množiny uživatelů komunit A, B, C vytvořené okolo klíčových slov z předmětů zpráv.

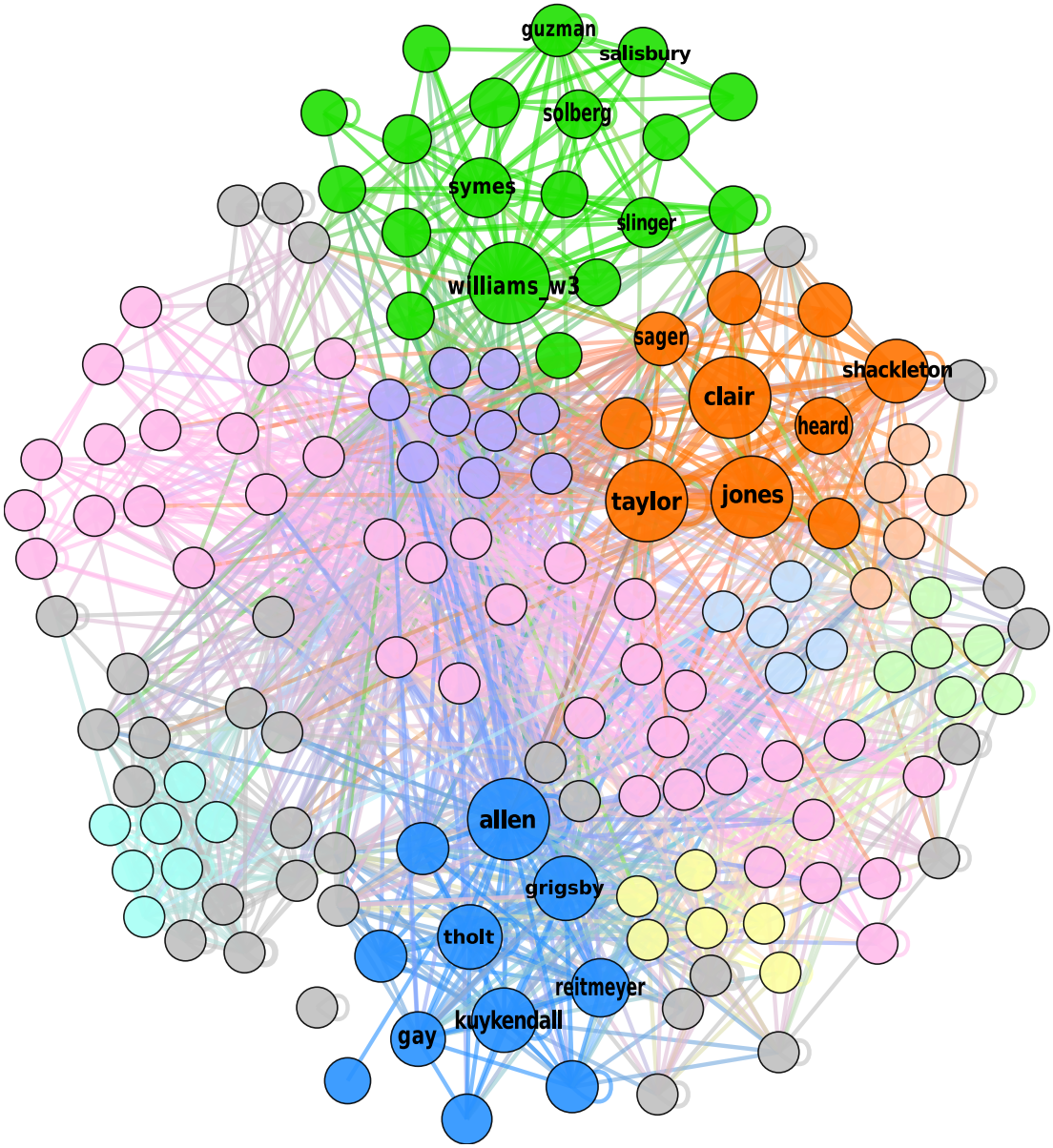
nejčastěji se vyskytujících slov v předmětech posílaných zpráv uvnitř jednotlivých komunit, dále jsme pro každou komunitu zvlášť prozkoumali předměty zpráv rozdělené do skupin podle dvojic uživatelů, mezi kterými byly zprávy posílány. Pomocí těchto seznamů jsme se pokusili najít nějaké prvky nebo zprávy, které by dokazovaly existenci komunity ve skutečném světě.

U všech tří vybraných komunit jsme se nejprve zaměřili slovo schůzka (*meeting*), protože uživatelé posílající si zprávy s tímto předmětem se buď osobně schůzek účastnili, nebo o nich ostatní informovali. V komunitách B a C (na Obrázku 4.1 modrá a oranžová) se slovo schůzka vyskytovalo v předmětech zpráv velkého množství uživatelů, o kterých můžeme říct, že tvoří jádro dané komunity. V komunitě A (zelená) tvoří množinu okolo slova schůzka pouze tři uživatelé, největší množinu uživatelů se nám podařilo vytvořit okolo slova oprávnit (*empower*).

V komunitě C (oranžová) se nám podařilo podložit nalezenou množinu uživatelů ještě i množinou odpovídající výskytům slova oběd (*lunch*). V komunitě B (modrá) jako další klíčové slovo posloužil plyn (*gas*), přičemž množiny vytvo-

řené okolo slov plyn a schůzka se liší pouze o jednu osobu. V případě komunity B lze proto dokonce říct, že to byla skupina lidí, která se zabývala prodejem a distribucí právě plynu.

Přehledy nalezených množin s příslušnými klíčovými slovy lze nalézt v Tabulce 4.4. Na obrázku (X) je potom k vidění graf reprezentující jádro komunity B (modrá na Obrázku 4.1) vytvořený na základě zpráv tvořených klíčovými slovy plyn a schůzka.



Obrázek 4.1: Vážený graf základních uživatelů s rozdělením získaným Lovaňským algoritmem s rezolucí nastavenou na 2.
V rozdělení je barevně zvýrazněno 10 největších komunit, dále jasnějšími barvami jsou zvýrazněny 3 komunity, u kterých jsme zkoumali reprezentativnost. V těchto třech komunitách jsou dále vrcholy škálovány podle B–centrality a navíc jsou vyznačeny jménem vrcholy s největšími hodnotami zkoumaných centralit (viz Tabulka 4.3).

Rozdělení s použitím randomizace

V této části bychom rádi prezentovali výsledky algoritmu s použitím randomizace při procházení vrcholů v prvním kroku Lovaňského algoritmu (Podkapitola 2.3.1). Pokus jsme prováděli na 10 různých semínkách (0, 16, 42, 83, 17, 121, 99, 24, 73, 104). Dále jsme brali v úvahu váhy hran a rezoluce byla nastavena na 2, stejně jako v rozdělení zvoleném k podrobnější analýze reprezentativnosti.

Jak vyplývá z porovnání v Tabulce 4.5, hodnoty získané nerandomizovanou verzí algoritmu se příliš neliší od průměrných hodnot získaných z verze randomizované. Proto můžeme nerandomizovanou verzi v tomto konkrétním případě považovat za poměrně dobrou aproximaci randomizovaných výsledků a v dalších pokusech budeme pracovat pouze s nerandomizovanou verzí algoritmu.

R	Mw	NCw	Sw	Vw	Diamw	Densw
2 + rand	0.65	27.9 $^{\pm 1.19}$	5.33 $^{\pm 0.23}$	45.56 $^{\pm 8.5}$	1.09 $^{\pm 0.02}$	0.81 $^{\pm 0.01}$
2	0.65	28	5.29	56.13	1.04	0.81

R–použitý rezoluční parametr, M–modularita,

NC–počet komunit, S–průměrná velikost komunity,

V–rozptyl velikosti komunit, w–započítáváme váhy hran,

Diam–průměrný průměr komunity, Dens–průměrná hustota komunity.

Tabulka 4.5: Porovnání rozdělení vrcholů na komunity získaného Lovaňským algoritmem s použitím a bez použití randomizace na neorientovaném grafu základních uživatelů. Poznámka: průměrné hodnoty získány z pokusů na 10 semínkách.

4.2.2 Výsledky získané na rozšířeném neorientovaném grafu

Rozšířený graf (Podkapitola 3.3.2) obsahuje kromě základních uživatelů i další uživatele, u kterých jsme zjistili, že komunikovali s velkým počtem jiných vrcholů. Rozšířený graf tedy celkem tvoří 244 vrcholů a 2975 hran. V průměru má každý vrchol 24 sousedů a vážený stupeň se pohybuje okolo hodnoty 903,98.

U rozšířeného grafu předpokládáme, že k velkým změnám, co se struktury komunit týče, nedojde. Základy větších komunit zůstanou stejné, komunity se pouze rozšíří o další uživatele, nebo připojí k sobě jiné menší komunity. Výsledky pokusů pro vybrané hodnoty rezolučního parametru jsou podobně jako pro graf základních uživatelů shrnuty v Tabulce 4.6.

Jak si lze všimnout, trend týkající se zvětšování rezoluce zkoumaný výše na grafu základních uživatelů je zde viditelný ještě zřetelněji. Zvlášť je viditelný na neohodnoceném grafu, kde se v tomto případě nevyskytují žádné zvláštní výkyvy. Výsledky se od předchozího grafu dále liší tím, jak k sobě konvergují hodnoty rozdělení na ohodnoceném a neohodnoceném grafu.

Nejlepší hodnoty modularity (0,29 pro neohodnocený graf a 0,64 pro ohodnocený) jsou nižší než hodnoty z rozdělení grafu základních uživatelů pouze o pár setin, což by mohlo být způsobeno větší velikostí grafu. Oproti tomu se výrazně snížil počet komunit (u obou grafů se trend ustálil na 16 komunitách). Kvůli zvětšení grafu došlo ke zvětšení průměrné velikostí komunit a vyšším hodnotám

R	M	Mw	NC	NCw	S	Sw
1	0.05	0.56	4	33	61	7.4
2	0.09	0.63	8	22	30.5	11.1
5	0.27	**0.64**	16	19	15.25	12.84
10	**0.29**	**0.64**	16	16	15.25	15.25
20	**0.29**	0.63	16	16	15.25	15.25

R	V	Vw	Diam	Diamw	Dens	Densw
1	8793.5	**81.51**	1.75	**1.4**	0.56	**0.71**
2	3824.75	184.54	1.88	1.5	0.57	0.68
5	398.94	287.08	1.81	1.37	0.65	0.68
10	359.3125	347.06	**1.69**	1.63	**0.64**	0.63
20	**352.06**	350.69	**1.69**	1.69	**0.64**	0.64

Poznámka: tučně zvýrazněny jsou nejlepší hodnoty.
R–použitý rezoluční parametr, M–modularita,
NC–počet komunit, S–průměrná velikost komunity,
V–rozptyl velikosti komunit, w–započítáváme váhy hran,
Diam–průměrný průměr komunity, Dens–průměrná hustota komunity.

Tabulka 4.6: Rozdělení vrcholů na komunity získaná Lovaňským algoritmem na neorientovaném rozšířeném grafu.

rozptylu. Zvětšení velikosti grafu vedlo i ke zvýšení průměru komunit a nižší hustotě (do grafu sice přibyly nové vrcholy, ale už ne takové množství nových hran).

Pro bližší prozkoumání komunit jsme vybrali rozdělení ohodnoceného grafu s rezolucí nastavenou na 5. Je to jedno z rozdělení, kde se pro ohodnocený graf podařilo dosáhnout nejvyšší hodnoty modularity a komunity jsou vnitřně poměrně hustě propojeny. Rozdělení nemá příliš vysoké hodnoty průměru ani rozptylu a zároveň se rozdělení na neohodnoceném grafu pro stejnou hodnotu rezolučního parametru začíná velmi blížit vyznačeným nejlepším hodnotám.

Je třeba ještě dodat, že rozdělení na neohodnoceném a ohodnoceném grafu se začínají výrazně podobat počtem i charakterem komunit od rezolučního parametru s hodnotou 10. Jak můžeme vidět i v Tabulce 4.7 porovnávající údaje pro jednotlivé komunity obou rozdělení, jsou tato rozdělení velice podobná (dvě třetiny komunit jsou stejné). Ale již pro rezoluční parametr s hodnotou 5 je jistá podobnost vidět.

Pokusili jsme se najít komunity odpovídající komunitám A, B, C z předchozího experimentu, na Obrázku 4.2 jsou značeny stejnými barvami jako v grafu základních uživatelů, tj. zelenou, modrou a oranžovou. Byly hledány podle významných vrcholů původních komunit. V Tabulce 4.8 můžeme vidět změny oproti původním komunitám ve velikosti a hustotě. K výraznému zvětšení došlo u komunity A (zelená), k největšímu rozšíření došlo pak u komunity C (oranžová), která má v rozšířeném grafu o 23 vrcholů více.

V Tabulce 4.9 jsou znázorněny významné vrcholy pro nové komunity. Oproti původním významným vrcholům (Tabulka 4.3) se zde objevilo několik nových jmen. V komunitách A (zelená) a C (oranžová) přibylo k původní množině významných vrcholů po jednom novém uživateli. V komunitě A je to vrchol, který patřil do původní komunity v grafu základních uživatelů, v komunitě C byl nový

vrchol přidán v rámci rozšíření. V komunitě B (modrá) došlo k mnohem většímu přeuspořádání, dva vrcholy zaujaly podřadnější pozice a důležitých se ujaly tři nové (dva z nich byly součástí komunity B v grafu základních uživatelů, jeden byl součástí jiné menší komunity).

Zajímavostí je, že oproti rozdělení na grafu základních uživatelů nejsou v tomto rozdělení tak výrazné rozdíly v B-centralitě v rámci jednotlivých komunit, což lze pozorovat i na grafu tohoto rozdělení Obrázek 4.2, ve kterém jsou

Id	S	Diam	Dens	Sw	Diamw	Densw
0	57	3	0.2	48	2	0.2
* 1	20	3	0.72	20	3	0.72
2	20	2	0.35	24	2	0.30
* 3	7	2	0.79	7	2	0.79
4	33	2	0.33	34	2	0.32
5	63	3	0.25	67	3	0.27
6	11	2	0.58	10	2	0.6
7	3	2	0.67	4	2	0.6
* 8	2	1	0.67	2	1	0.67
* 9	10	2	0.78	10	2	0.78
* 10	6	2	0.43	6	2	0.43
* 11	1	0	1	1	0	1
* 12	5	1	0.87	5	1	0.87
* 13	1	0	1	1	0	1
* 14	4	2	0.6	4	2	0.6
* 15	1	0	1	1	0	1

Poznámka: * jsou označeny ID komunit,
pro která se obě rozdělení shodují.
Id–identifikátor komunity, S–velikost komunity,
Diam–průměr komunity, Dens–hustota komunity,
Sw, Diamw, Densw–velikost, průměr a hustota
komunity v ohodnoceném grafu.

Tabulka 4.7: Porovnání rozdělení na ohodnoceném a neohodnoceném grafu s rezolučním parametrem nastaveným na 10.

Komunita	V	E	D	Vw	Ew	Dw
A	19	70	0.37	24	91	0.3
B	11	44	0.67	12	62	0.79
C	10	41	0.75	33	186	0.33

V–počet vrcholů, E–počet hran, D–hustota,
w–rozdělení na ohodnoceném rozšířeném grafu.

Tabulka 4.8: Porovnání velikosti a hustoty vybraných komunit z rozdělení na ohodnoceném grafu základních uživatelů (pro rezoluční parametr s hodnotou 2) a na ohodnoceném rozšířeném grafu (pro rezoluční parametr s hodnotou 5).

Centralita	Uživatelé z komunity A
D–centralita	williams_w3, symes, slinger, guzman, solberg
B–centralita	williams_w3, slinger, symes, salisbury, semperger
C–centralita	williams_w3, symes, slinger, guzman, salisbury
Centralita	Uživatelé z komunity B
D–centralita	grigsby, allen, tholt, sanchez, kuykendall
B–centralita	sanchez, grigsby, south, ermis, allen
C–centralita	grigsby, allen, tholt, sanchez, kyukendall
Centralita	Uživatelé z komunity C
D–centralita	jones, taylor, shackleton, heard, clair
B–centralita	jones, taylor, clair, sager, sayre
C–centralita	jones, taylor, shackleton, heard, clair

Tabulka 4.9: Vrcholy s nejvyšší hodnotou centralit pro vybrané komunity v rozšířeném ohodnoceném grafu s rezolučním parametrem 5.

vybrané komunity škálovány dle B-centrality podobně jako v grafu rozdělení na základních uživatelích (Obrázek 4.1). Podobné závěry, i když ne tak značné, lze vyvodit i pro zbylé pozorované centrality.

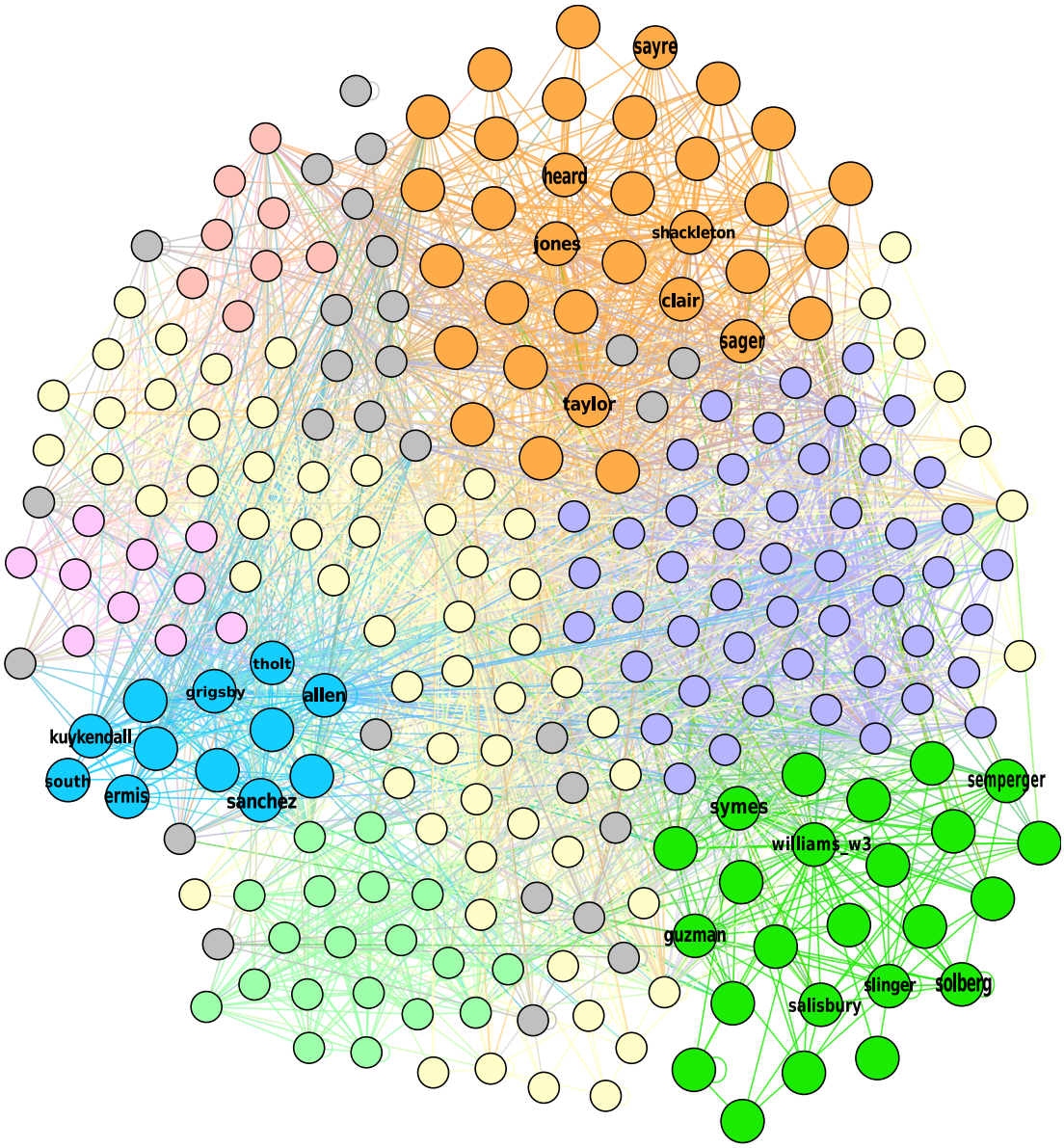
4.2.3 Výsledky získané na základním a rozšířeném orientovaném grafu

Orientované grafy mají oproti neorientovaným více hran (2094 pro graf základních uživatelů, 3623 pro rozšířený graf) a nižší stupně vrcholů, protože musíme rozlišovat mezi vstupním a výstupním stupněm. Také lze předpokládat, že se budou komunity mírně lišit, protože některá spojení mezi vrcholy v orientovaném grafu nemusí být oboustranná, nebo naopak budou nalezené komunity o něco jednoznačnější.

Porovnání výsledků s výsledky na neorientovaných grafech jsou vidět v Tabulce 4.10. Přesto, že se příliš nezměnila modularita rozdělení, došlo k razantnímu zvýšení celkového počtu detekovaných komunit, což vedlo ke snížení průměrné velikosti a rozptylu a ke zlepšení průměru a hustoty komunit.

Přesto lze v grafech nelézt obdoby označených komunit A, B, C. V grafu základních uživatelů jsou mírně menší, komunita A je dokonce rozdělená na dvě. Pro rozšířený graf platí opačný trend. Převládají zde velké komunity, které jsou proloženy několika extrémně malými o 2 nebo 3 vrcholech. Sledované komunity A, B, C jsou tedy o něco větší než jim odpovídající v neorientovaném grafu.

Protože neorientovaný graf lze reprezentovat jako orientovaný tak, že za každou hranu přidáme dvě orientované v opačných směrech, tak z výsledků na orientovaných grafech můžeme usoudit, že komunikace mezi mnoha uživateli probíhala oboustranně a většinou vyváženě. Sice došlo k některým strukturálním změnám, ale jádra většiny velkých komunit zůstala v orientovaných grafech zachována.



Obrázek 4.2: Vážený rozšířený graf s rozdělením získaným Lovaňským algoritmem s rezolucí nastavenou na 5.
V rozdělení je barevně označeno 8 největších komunit, dále jasnějšími barvami jsou zvýrazněny podrobněji zkoumané 3 komunity. V těchto třech komunitách jsou dále vrcholy škálovány podle B–centrality a navíc jsou jménem vyznačeny vrcholy s největšími hodnotami zkoumaných centralit (viz Tabulka 4.9).

Orientovaný							
Graph	R	M	NC	S	V	Diam	Dens
basic	2	0.24	32	4.63	44.92	1.13	0.78
ext	5	0.28	26	9.38	217.85	1.38	0.67
Neorientovaný							
Graph	R	M	NC	S	V	Diam	Dens
basic	2	0.25	19	7.79	147.32	1.53	0.72
ext	5	0.27	16	15.25	398.94	1.81	0.65

Graph–typ grafu, R–použitý rezoluční parametr,
M–modularita, NC–počet komunit,
S–průměrná velikost komunity, V–rozptyl velikosti
komunit, Diam–průměrný průměr komunity,
Dens–průměrná hustota komunity.

Tabulka 4.10: Porovnání rozdělení vrcholů na komunity získaných Lovaňským algoritmem na orientovaném a neorientovaném grafu základních uživatelů.

4.2.4 Závěrečné zhodnocení algoritmu

Algoritmus je schopný najít poměrně jednoznačné a rovnoměrné (až na pár extrémně malých komunit) rozdělení sítě na reprezentativní komunity. Jeho výhodou je snadná implementace, široké uplatnění a univerzálnost – lze ho poměrně jednoduše rozšířit na orientované grafy (Vzorec 2.11). Zároveň má ale algoritmus tendenci slučovat mnoho vrcholů do jedné velké komunity, což vedlo ve většině získaných rozdělení ke vzniku jedné výrazně větší centrální komunity.

Sice je algoritmus prezentován jeho hierarchický, na datech z Enronu se nám nicméně žádnou hierarchickou strukturu odhalit nepodařilo. Z výsledků experimentů můžeme dále usoudit, že váhy hran hrají v mnoha sítích důležitou roli a mohou pro určitá nastavení parametrů výrazně ovlivnit konečnou podobu komunit, viz Tabulka 4.2. Vážené hrany také způsobují vyšší hodnotu modularity rozdělení.

Zároveň jsme zjistili, že orientace hran sice příliš neovlivní celkovou modularitu rozdělení, ale má vliv na podobu komunit a jejich počet. Započítávání hran a jejich orientace vede k získání přesnějších výsledků, které lépe reflektují skutečnou podobu sítě než její neorientovaná varianta.

Komunity, které jsme zkoumali blíže při zjišťování reprezentativnosti, a jejich klíčoví uživatelé nám poslouží jako základní opěrné body při zkoumání reprezentativnosti rozdělení vrcholů na komunity nalezných dalšími algoritmy prezentovanými níže.

4.3 Experimenty s algoritmem SCAN

Jak již bylo popsáno v předchozím textu, SCAN (Podkapitola 2.3.2) je algoritmem, který bere v úvahu pouze strukturu sítě. Pro každou dvojici sousedních vrcholů spočítá jejich podobnost a do množiny podobných sousedů (ε-okolí) každého vrcholu přiřadi pouze takové sousedy, se kterými je dostatečně podobný. Mi-

nimální podobnost je určena parametrem ε. Jako jádro potom označíme vrchol, který má minimálně μ vrcholů ve svém ε-okolí. Komunity jsou potom sjednocením sousedících okolí různých jader.

Specialitou tohoto algoritmu je schopnost určovat rozcestníky a odlehlé vrcholy. Ty vzniknou z takových vrcholů, které jsme nepřiřadili k žádné komunitě, tj. neležely v okolí žádného jádra.

Algoritmus má tedy dva parametry. ε definuje minimální podobnost dvojice vrcholů pro jejich přiřazení do ε-okolí každého z nich. μ určuje minimální velikost ε-okolí jádra, jinými slovy také minimální velikost komunity v síti. Autoři DSCAN [14] uvádějí, že po provedení experimentů s algoritmem SCAN na různých grafech se optimální hodnoty ε nacházely v intervalu od 0,4 do 0,8. Proto se v experimentech zaměříme na ε právě z tohoto intervalu.

Pro porovnání výsledků použijeme stejné prostředky jako v experimentech s Lovaňským algoritmem (Podkapitola 4.2): celkovou modularitu rozdělení, počet komunit, průměr a rozptyl velikostí komunit a hustotu a průměr komunit. Pro určení reprezentativnosti komunit budeme vycházet z již ověřených množin z experimentů s Lovaňským algoritmem (Tabulka 4.4). Pokud se nám ale nepodaří nalézt komunity odpovídající těmto množinám, použijeme k analýze grafy zobrazující komunikaci uživatelů v různou denní dobu a předměty posílaných zpráv mezi uživateli.

Je třeba také zmínit, že kvůli charakteristice algoritmu a způsobu, kterým je definována podobnost (Vzorec 2.12), nemá smysl algoritmus aplikovat v původní podobě na ohodnocené grafy. Dále v článku [15] nebyl uveden ani žádný způsob rozšíření algoritmu na orientované grafy. Proto se v experimentech zaměříme pouze na neohodnocené neorientované grafy.

4.3.1 Výsledky získané na neorientovaných grafech

Výsledky na množině základních uživatelů

Jako první představíme výsledky na grafu základních uživatelů. Tvoří ho 148 vrcholů a 1535 hran, průměrný počet sousedů v grafu se blíží 21 a průměrný vážený stupeň má hodnotu 734,88.

SCAN funguje relativně dobře, pokud se podaří najít optimální hodnoty parametrů ε a μ. Není však známý žádný univerzální způsob, jak tyto hodnoty najít. Abychom nemuseli zkoušet každou možnou dvojici parametrů, zjednodušili jsme si práci způsobem popsaným v následujícím odstavci.

Hlavním cílem bylo předejít tvorbě nadměrného počtu jader s velkým ε-okolím. Velký počet jader totiž vedl k tomu, že se okolí téměř všech jader překrývala, což vedlo k formování pouze jedné velké komunity. Pro každou zkoumanou hodnotu ε jsme při prvním pokusu spočítali průměrnou velikost ε-okolí a parametr μ jsme nastavovali blízký této průměrné velikosti. Jak jsme si později při experimentech ověřili, hodnoty μ výrazně větší, nebo menší než průměrná velikost ε-okolí způsobovaly horší výsledky, konkrétně vedly ke vzniku většího počtu malých komunit a vysokého počtu odlehlých vrcholů a rozcestníků.

Ze všech provedených experimentů jsme se snažili vybrat ty, kde počet rozcestníků a odlehlých vrcholů byl co nejmenší. Nejlepších výsledků se nám podařilo dosáhnout pro ε z intervalu od 0,6 do 0,65 a μ rovné 2, nebo 3. Konkrétní výsledky lze pozorovat v Tabulce 4.11.

	ε	μ	M	NC	S	V	O	H	Diam	Dens
*	0.60	2	0.16	7	16	528	4	32	2.29	0.67
	0.60	3	0.16	7	15.14	501.36	10	32	2	0.53
	0.61	2	0.18	8	13.38	317.23	7	34	2	0.7
	0.61	3	0.18	9	11.22	250.84	8	39	1.78	0.71
	0.62	2	0.18	9	11.67	284	7	36	1.89	0.74
	0.62	3	0.18	9	10.89	243.43	9	41	1.78	0.72
	0.63	2	0.18	11	9.36	209.14	6	39	1.81	0.72
	0.64	2	0.21	11	9	121.27	4	45	1.9	0.75
	0.64	3	0.21	10	9	101.8	8	50	1.8	0.76
	0.65	2	**0.24**	13	7.5	46.09	4	47	**1.7**	**0.79**
	0.65	3	0.23	10	8.4	**37.64**	9	55	**1.7**	0.78
	0.66	2	0.22	12	7.33	38.39	6	54	1.83	0.75

Poznámka: 1) tučně jsou zvýrazněny nejlepší hodnoty,
2) řádek s nejmenším součtem rozcestníků a odlehlých vrcholů označen *.
ε–hranice podobnosti, μ–minimální velikost okolí jádra, M–modularita,
NC–počet komunit, S–průměrná velikost komunity, V–rozptyl velikosti komunit
Diam–průměrný průměr komunity, Dens–průměrná hustota komunity,
O–počet odlehlých vrcholů, H–počet rozcestníků.

Tabulka 4.11: Rozdělení získaná algoritmem SCAN na neorientovaném neohodnoceném grafu základních uživatelů.

Z Tabulky 4.11 je patrné, že optimální nastavení parametrů pro tento graf jsou $\varepsilon = 0.65$ a $\mu = 2$. Odpovídající rozdělení má největší modularitu, nejvyšší hustotu komunit i nejmenší rozptyl v jejich velikostech. V Tabulce 4.11 lze vidět i postupné zlepšování sledovaných parametrů pro ε blížící se k 0,65 zleva a jejich zhoršování po překročení optimálních hodnot.

Více než třetinu všech vrcholů optimálního rozdělení ale tvoří rozcestníky a odlehlé vrcholy. Bohužel se nám nepodařilo jejich počet výrazně snížit, zvýšení hodnoty ε vedlo k ještě většímu počtu rozcestníků a odlehlých vrcholů, snížení naopak k formování jedné obrovské komunity. Množiny odlehlých vrcholů i rozcestníků ale ve většině rozdělení z Tabulky 4.11 tvořily přibližně stejné vrcholy. Uživatelé *neal*, *fisher* a *merris* se vyskytovali jako odlehlé vrcholy v každém ze zkoumaných rozdělení. Častými rozcestníky byli například uživatelé *zufferli*, *tycholiz*, *arora*, *hain*, *germany* a *pimenov*.

Výsledky na rozšířené množině

Nyní se podíváme na rozšířený graf, který tvoří 244 vrcholů a 2975 hran. Průměrný počet sousedů jednoho vrcholu se pohybuje okolo 24 a vážený stupeň okolo 903.98. Předpokládáme, že se výsledky nebudou příliš lišit od základního grafu podobně jako u Lovaňského algoritmu a ani se příliš nezmění hodnoty optimálních parametrů. Konkrétní výsledky lze vidět v Tabulce 4.12.

Jak si můžeme všimnout, na větším grafu si algoritmus vedl o něco hůře. Nejvyšší hodnota modularity (0,13) je skoro o polovinu menší než v případě grafu základních uživatelů (0,24). Oproti předchozímu grafu máme hned 3 optima s nejvyšší modularitou, která se výrazně liší pouze v rozptylu velikostí komunit.

	ε	μ	M	NC	S	V	O	H	Diam	Dens
*	0.57	2	0.11	14	10.93	408.92	**27**	**64**	1.86	0.63
	0.57	3	0.11	8	16.88	592.11	38	71	2.25	0.60
	0.58	4	**0.13**	8	13.63	287.23	16	119	2.0	0.74
	0.58	5	0.12	6	16	342.33	20	128	2.17	0.72
	0.59	4	**0.13**	7	14.14	270.7	20	125	2.14	0.72
	0.59	5	0.12	6	15.33	289.89	23	129	2.17	0.73
	0.60	3	0.12	9	11.44	232.91	28	113	2.11	0.66
	0.60	4	0.12	7	13	233.14	30	123	2.14	0.72
	0.61	2	0.12	14	8	157.57	26	106	1.71	0.68
	0.61	3	**0.13**	8	11.38	78.48	29	124	2.13	0.72
	0.62	2	0.12	16	6.5	56.38	24	116	1.56	0.71
	0.62	3	0.12	8	10.5	79.25	33	127	2.0	0.72
	0.63	2	0.11	16	6.06	47.31	24	123	1.56	0.72
	0.63	3	0.10	8	9.75	66.69	38	128	2.0	0.71
	0.64	3	0.10	8	8.75	54.69	36	138	1.63	0.83
	0.65	2	0.10	15	5.33	**25.69**	19	145	**1.33**	**0.86**
	0.65	3	0.10	8	8	29.0	51	129	1.63	0.84

Poznámka: 1) tučně zvýrazněny nejlepší hodnoty,
2) řádek s nejmenším součtem rozcestníků a odlehlých vrcholů označen *.
ε–hranice podobnosti, μ–minimální velikost okolí jádra, M–modularita,
NC–počet komunit, S–průměrná velikost komunity, V–rozptyl velikosti komunit
Diam–průměrný průměr komunity, Dens–průměrná hustota komunity,
O–počet odlehlých vrcholů, H–počet rozcestníků.

Tabulka 4.12: Rozdělení získaná algoritmem SCAN na neorientovaném neohodnoceném rozšířeném grafu.

Když se podíváme na další parametry, tak se v nich zachovává stejný trend jako v Tabulce 4.11. Pro nižší ε máme nízký počet odlehlých vrcholů a rozcestníků, se zvyšujícím se ε se pak snižuje rozptyl velikostí, průměr komunit a zvyšuje se hustota. Rozdělení v dolní části tabulky se i přes skvělé hodnoty více parametrů nehodí jako dobrá reprezentace komunitní struktury sítě, především kvůli nadměrnému počtu odlehlých vrcholů a rozcestníků (tvoří až tři čtvrtiny všech vrcholů v grafu). Podobně jako v případě grafu základních uživatelů ale můžeme potvrdit, že se mnohé množiny odlehlých vrcholů a rozcestníků prolínají. Stálou množinu odlehlých vrcholů například tvoří uživatelé *wright*, *petrochko*, *linnell* a *fisher*.

K bližšímu prozkoumání jsme zvolili dvě z rozdělení s nejvyšší modularitou, a to s $\varepsilon = 0.58$, $\mu = 4$ a $\varepsilon = 0.31$, $\mu = 3$. Rozcestníky a odlehlé vrcholy sice stále tvoří okolo poloviny všech vrcholů grafu, jsou to ale jedny z nejmenších naměřených hodnot. Navíc hustota nalezených komunit je poměrně vysoká, z čehož usuzujeme, že by se nalezené rozdělení na komunity mohlo blížit skutečné struktuře sítě.

Reprezentativnost rozdělení

Reprezentativnost komunit pro graf základních uživatelů budeme zkoumat na výše zmiňovaném optimálním rozdělení s parametry $\varepsilon = 0.65$, $\mu = 2$. Pro rozšířený graf porovnáme výsledky dvou vybraných rozdělení na komunity ($\varepsilon = 0.58$, $\mu = 4$ a $\varepsilon = 0.31$, $\mu = 3$).

Protože jsme si již ověřili reprezentativnost některých komunit pro Lovaňský algoritmus (Tabulka 4.4), pokusíme se najít důležité vrcholy z těchto komunit a podívat se, zda netvoří komunity i v rozdělení nalezeném algoritmem SCAN.

V rozdělení pro graf základních uživatelů jsme schopni nalézt komunity A, B, C z Tabulky 4.4 (na Obrázku 4.4 jsou opět značeny zelenou, modrou a oranžovou barvou). V tomto případě má A 10, B 11 a C 8 vrcholů. Počet vrcholů zůstal zachován jenom pro komunitu B, A se zmenšila o polovinu a C o 2 vrcholy. V komunitě C (oranžová) chybí například uživatel *sager*. Ke komunitě A (zelená) by mohla patřit další komunita obsahující pouze 3 vrcholy ale nacházející se v těsném sousedství s A (značena světle zelenou barvou). Na Obrázku 4.4 je k nahlédnutí kompletní rozložení na komunity na grafu základních uživatelů.

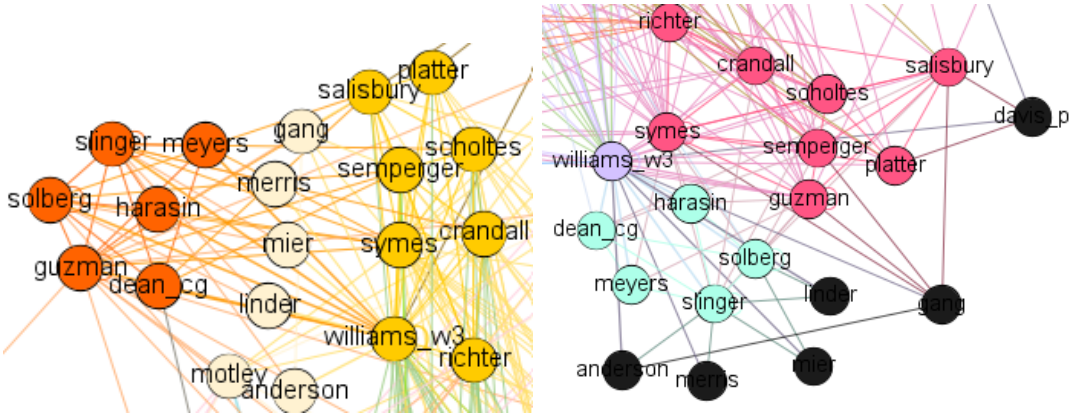
Pro rozšířený graf jsme vybrali dvě rozdělení ($\varepsilon = 0.58$, $\mu = 4$ a $\varepsilon = 0.31$, $\mu = 3$). Mají nejvyšší naměřenou modularitu na rozšířeném grafu a dokonce i stejný počet detekovaných komunit. Opět jsme se pokusili vyhledat komunity z Tabulky 4.4. V obou grafech byla jasně určená komunita B čítající 11 vrcholů. Komunita C v prvním rozdělení ($\varepsilon = 0.58$, $\mu = 4$) má 11 vrcholů, v druhém ($\varepsilon = 0.31$, $\mu = 3$) má 8 vrcholů. Při bližším pohledu ale zjistíme, že by ke komunitě mohly patřit i dva odlehlé vrcholy, které mají sousedy pouze v komunitě C.

Největší změny jsou patrné na komunitě A. V obou rozděleních jsou původní klíčoví uživatelé rozdělení na dvě různé komunity, které se napříč oběma rozděleními neshodují, viz Obrázek 4.3. V prvním rozdělení ($\varepsilon = 0.58$, $\mu = 4$) spojuje obě komunity (oranžová a žlutá) 6 rozcestníků, které mají sousedy pouze ve zmíněných dvou komunitách. Ve druhém rozdělení ($\varepsilon = 0.31$, $\mu = 3$) je jeden z klíčových uživatelů – *williams_w3* – označen jako rozcestník. To skutečně odpovídá jeho roli v sítí, protože je jedním z mála vrcholů z původní komunity A, které mají spojení s vrcholy mimo tuto komunitu. Kromě tohoto rozcestníku se dá ke dvojici komunit (tmavě růžová a světle tyrkysová) připojit i několik odlehlých vrcholů (značeny černou barvou), které mají většinou jednoho, nebo dva sousedy v některé z komunit.

4.3.2 Závěrečné zhodnocení algoritmu

Podařilo se nám najít optimální hodnoty parametrů pro oba zkoumané grafy a zjistili jsme, že nalezené komunity by mohly odpovídat komunitám ve skutečném světě. Některé nalezené komunity odpovídaly těm detekovaným Lovaňským algoritmem a celková modularita rozdělení získaného algoritmem SCAN byla v porovnání s rozděleními získanými Lovaňským algoritmem o více než jednu desetinu menší, jak jsme původně předpokládali.

Jak se ukázalo především u rozšířeného grafu, schopnost nalézt rozcestníky a odlehlé vrcholy může být užitečná. Jako příklad lze vzít vrchol *williams_w3* v rozdělení ($\varepsilon = 0.31$, $\mu = 3$), který podle Lovaňského algoritmu patřil do sjednocení dvou komunit na Obrázku 4.3, ale SCAN ho označil jako rozcestník. Vrchol



Obrázek 4.3: Částí rozdělení pro rozšířený graf, které reprezentují komunitu A z předchozích pokusů. Vlevo rozdělení ($\varepsilon = 0.58$, $\mu = 4$), vpravo ($\varepsilon = 0.31$, $\mu = 3$).

zprostředkovává komunikaci komunity A se zbytkem sítě, proto pravděpodobně byl rozcestníkem i v skutečném světě.

Napříjemnými vlastnostmi algoritmu jsou nutnost manuálního hledání optimálních parametrů a nadměrná produkce rozcestníků, které tvoří v některých rozděleních i více než tři čtvrtiny všech vrcholů. Na druhou stranu je ale snadno implementovatelný, rychlý a jak bylo popsáno výše v textu, právě tento algoritmus budeme rozšiřovat pro zkoumání dynamické sítě (Podkapitola 2.3.4).

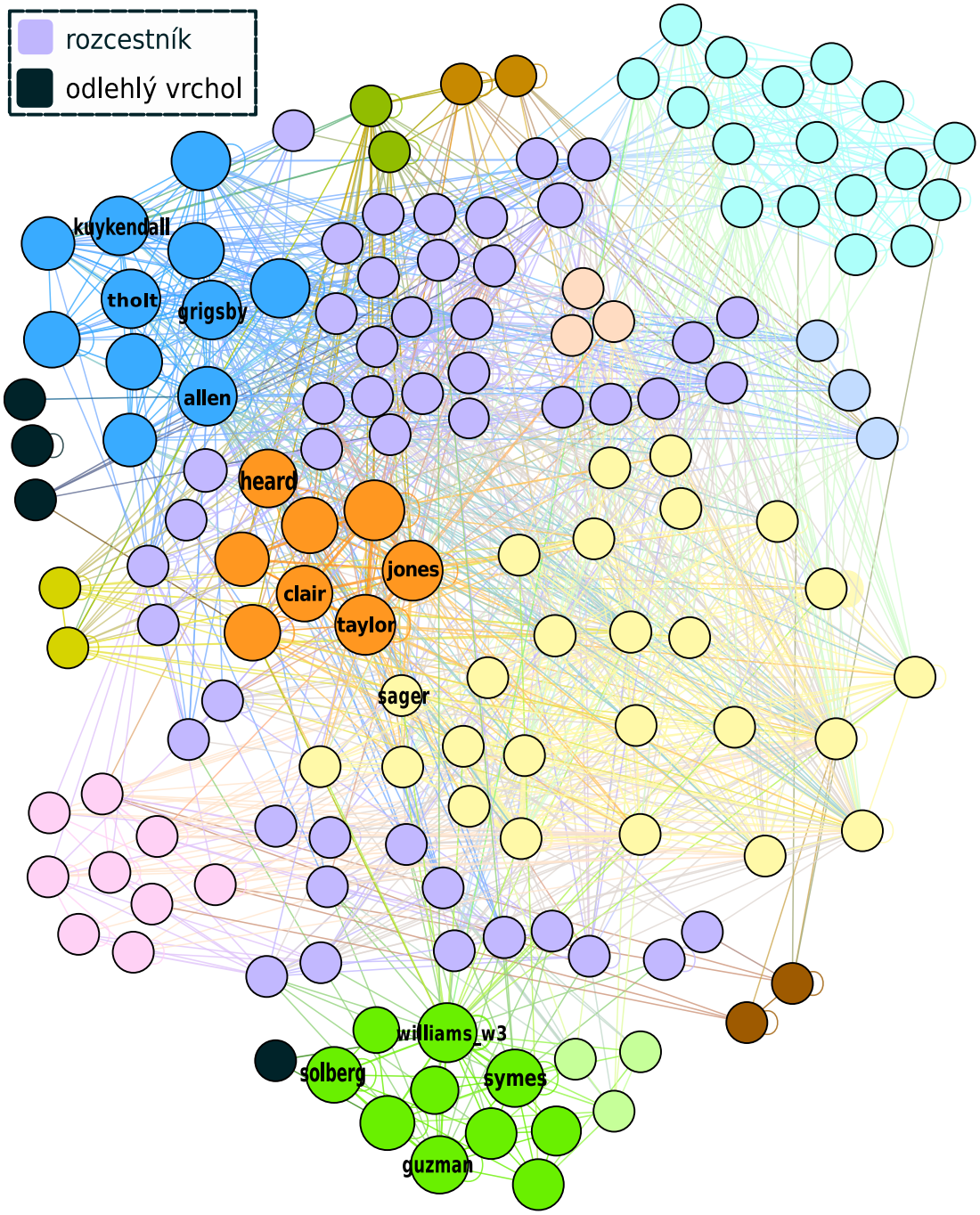
4.4 Experimenty s algoritmem klastrování pomocí hran

Jak jsme již popsali v Podkapitole 2.3.3, klastrování pomocí hran spojuje do komunit hrany. Rozdělení hran na komunity způsobí vznik rozdělení na vrcholech, kde se komunity vzájemně překrývají. Překryvy nám pomůžou odhalit vrcholy, které by mohly mít v síti roli spojnic mezi různými skupinami.

Klastrování pomocí hran je iterativním algoritmem, tedy výstupem je dendrogram rozdělení hran na komunity. Jako optimální bereme rozdělení s největší hustotou (Vzorec 2.14). V některých grafech může docházet k tomu, že se hrany budou spojovat do příliš velkých komunit. Abychom tomu zamezili, lze omezit minimální podobnost dvou hran a hrany s podobností menší než nastavený limit nebudou, když na ně přijde řada, sloučeny do jedné komunity.

Pro interpretaci výsledků používáme tři rozdělení na komunity. Prvním je rozdělení hran, které dostaneme na výstupu algoritmu, měříme na něm hustotu, průměr a hodnoty centralit vrcholů pro komunity. Další rozdělení, tentokrát vrcholové, je vytvořené z předchozího hranového, každému vrcholu je v něm přiřazena množina komunit, do kterých patří hrany incidentní s daným vrcholem. Toto rozdělení nám slouží ke změření všech metrik spojených s velikosti komunit.

Poslední rozdělení je opět vrcholové, ale každému vrcholu je v něm přiřazena právě jedna komunita, která je nejvíce zastoupená mezi hranami incidentními s daným vrcholem. Toto rozdělení použijeme na změření modularity, na rozdělení s překrývajícími se komunitami modularitu totiž měřit nelze.



Obrázek 4.4: Graf základních uživatelů s rozdělením získaným algoritmem SCAN. V rozdělení jsou jasnějšími barvami (oranžová, modrá, zelená) zvýrazněny podrobněji zkoumané 3 komunity. V těchto třech komunitách jsou dále vrcholy škálovány podle D–centrality a navíc jsou jménem vyznačeny významné vrcholy, podle kterých jsme komunity identifikovali.

Kromě popsaných změn přidáme několik nových parametrů: procento vrcholů v překryvech a průměrný počet komunit, do kterých patří jeden vrchol. Navíc předpokládáme, že počet nalezených komunit bude o něco větší než u Lovaňského algoritmu a SCAN kvůli umožněným překryvům. Zároveň ale očekávame menší variabilitu ve velikosti a hustotě nalezených komunit.

4.4.1 Výsledky získané na neorientovaných grafech

Opět algoritmus aplikujeme na dva grafy — graf základních uživatelů tvořený 148 vrcholy a 1535 hranami a na rozšířený graf tvořený 244 vrcholy a 2975 hranami. Průměrný počet sousedů jednoho vrcholu prvního je 21 a průměrný vážený stupeň se blíží 734,88. Pro druhý graf je průměrný počet sousedů 24 a vážený stupeň se pohybuje okolo 903,98.

V experimentech nebylo třeba použít omezení pro podobnost hran, které by zvýšilo počet detekovaných komunit, jejich počet totiž i tak vyšel dost vysoký (500 u základního grafu, 1166 u rozšířeného grafu pro nejlepší rozdělení podle hustoty). Po zvýšení hranice minimální podobnosti pouze více vzrostl.

Podrobné údaje pro jednotlivé grafy lze najít v Tabulkách 4.13 a 4.14. Nejlepší rozdělení dle hustoty (Vzorec 2.14) má v případě grafu základních uživatelů zároveň i nejvyšší modularitu pro zjednodušené vrcholové rozdělení popisované výše. V rozděleních z nižších a vyšších úrovní dendrogramu dále dochází k postupnému poklesu modularity. V rozšířeném grafu se nejlepší modularity podařilo dosáhnout na úrovni 250, oproti tomu nejlepší rozdělení dle hustoty se nachází na úrovní 120.

level	M	NC	S	V	NO	AvNO	Diam	Dens
10	0.07	1443	1.99	**0.27**	0.02	1.06	**0.95**	**0.99**
50	0.12	1021	2.29	1.4	0.17	1.5	1.1	0.92
90	0.20	544	2.90	7.14	0.31	2.82	1.3	0.84
* 98	**0.22**	500	2.99	8.19	0.32	3.07	1.3	0.83
120	0.16	375	3.21	16.57	0.34	4.09	1.34	0.82
200	0.01	96	4.23	192.78	0.35	15.99	1.32	0.83

Poznámka: 1) tučně zvýrazněny jsou nejlepší hodnoty,

2) * je zvýrazněna úroveň s nejvyšší hustotou rozdělení.

level–vrstva dendrogramu, NC–počet komunit,

S–průměrná velikost komunity, V–rozptyl velikosti komunit,

Diam–průměrný průměr komunity, Dens–průměrná hustota komunity,

NO–procento vrcholů v překryvech, AvNO–průměrný počet komunit na vrchol.

Tabulka 4.13: Rozdělení získaná klastrováním pomocí hran na neorientovaném neohodnoceném grafu základních uživatelů.

Dalším pozorováním, které můžeme provést na základě tabulek, jsou nízké hodnoty průměrné velikosti komunit, rozptyl velikostí a také průměr komunit. Zároveň si můžeme všimnout i velmi vysoké hustoty (nad 80 % pro základní i rozšířený graf). Vše výše popsané je pravděpodobně způsobené extrémně vysokým počtem komunit, z nichž většina má minimální velikost, a to 2–3 vrcholy.

level	M	NC	S	V	NO	AvNO	Diam	Dens
10	0.04	2854	2	**0.25**	0.01	1.04	**0.97**	**1**
50	0.06	2448	2.14	0.52	0.12	1.22	1.08	0.95
100	0.12	1440	2.66	4.12	0.27	2.07	1.26	0.87
* 120	0.13	1166	2.88	6.58	0.3	2.55	1.29	0.85
200	0.14	633	3.37	24.93	0.38	4.7	1.41	0.8
250	**0.15**	400	3.72	69.09	0.37	7.44	1.41	0.8
300	0.06	296	3.89	138.16	0.36	10.05	1.35	0.81

Poznámka: 1) tučně zvýrazněny jsou nejlepší hodnoty,

2) * je zvýrazněna úroveň s nejvyšší hustotou rozdělení.

level–vrstva dendrogramu, NC–počet komunit,

S–průměrná velikost komunity, V–rozptyl velikosti komunit,

Diam–průměrný průměr komunity, Dens–průměrná hustota komunity,

NO–procento vrcholů v překryvech, AvNO–průměrný počet komunit na vrchol.

Tabulka 4.14: Rozdělení získaná klastrováním pomocí hran na neorientovaném neohodnoceném rozšířeném grafu.

Reprezentativnost rozdělení

Reprezentativnost budeme v případě grafu základních uživatelů zkoumat pomocí již ověřených komunit (Tabulka 4.4). Pokud takové nenalezneme, přistoupíme k analýze pomocí předmětů posílaných zpráv a grafů denních dob (Podkapitola 3.5).

Komunity nalezené algoritmem lze rozdělit do dvou skupin. Komunity z první skupiny se podobají těm, které jsme detekovali Lovaňským algoritmem a algoritmem SCAN, včetně podrobně zkoumaných komunit A, B, C. Na Obrázku 4.5 je znázorněno 5 nejvýznamnějších hranových komunit na grafu základních uživatelů, které odpovídají komunitám z rozdělení získaných Lovaňským algoritmem a SCAN. Komunity A a B jsou značeny zelenou a béžovou barvou. V komunitě C jsou vrcholy obarvené dvěma barvami (tyrkysová, žlutá) z toho důvodu, že žluté vrcholy patří větší měrou do jiné hranové komunity než vyznačená hranová komunita C.

Na stejném obrázku v dolní části je znázorněn druhý typ komunit, který odpovídá množinám hran, které tvoří spojení mezi dvojicemi komunit prvního typu. Komunity druhého typu jsou převážně bipartitními grafy. Reprezentativnost druhého typu komunit je třeba ověřit pomocí hledání společných předmětů zpráv, protože neexistují žádné jim odpovídající komunity v rozdělení zkoumaných dříve.

Jako první prozkoumáme komunitu spojující B (béžová) s centrální komunitou (vrcholy značeny růžovou barvou), na obrázku jsou hrany zkoumané komunity značeny tmavě červenou barvou. Z předmětů zpráv se nic zjistit nepodařilo, většina předmětů byla totiž prázdná. Jediné, co víme, že se uživatel *dasovich* účastnil schůzek s uživateli *grigsby* a *tholt* a také, že se scházeli uživatelé *grigsby* a *kean*.

Další komunita druhého typu spojuje na obrázku modrou komunitu s centrální, její hrany jsme označili tmavě modrou barvou. Zde již množství prázdných předmětů bylo menší, přesto se nám nepodařilo najít žádná klíčová slova, která by spojovala většinu vrcholů v komunitě. Detekovaná komunita je tvořená spíše několika množinami hran, které spojují sousedy jednoho konkrétního vrcholu.

U poslední analyzované komunity druhého typu (hrany značeny zelenou barvou) jsme dosáhli podobně neúspěšných výsledků jako u předchozích dvou. Z toho usuzujeme, že druhý typ komunit není podložen žádnými existujícími komunitami ve skutečném světě a pouze seskupuje do množin konverzace mezi komunitami prvního typu. Stejný závěr můžeme učinit i pro rozšířený graf.

V rozšířeném grafu jsou ze známých komunit zachovány B (žlutá) a C (tyrkysová). Jak je vidět na Obrázku 4.6, vrcholy z komunity A jsou rozděleny na dvě menší (fialová a oranžová). V první se v předmětech zpráv objevovaly převážně informace o převodu peněz, takže by mohla tvořit i v reálném světě podkomunitu původní A. V druhé všechny poslané zprávy obsahovaly pouze prázdný předmět, proto se nedá rozhodnout, zda tato komunita má nějaký základ v skutečném světě.

Dále zanalyzujeme komunitu, která je na Obrázku 4.6 značena zeleně. Neodpovídá totiž žádnému z výše představených typů, mohla by ale tvořit skutečnou komunitu. Nejčastějšími slovy napříč předměty zpráv jsou Kalifornie (*california*), energie (*energy*), schůzka (*meeting*), síla (*power*). Tato slova se často vyskytují v konverzacích mezi většinou dvojic uživatelů. Z toho usuzujeme, že se tato skupina lidí pravděpodobně zabývala distribucí a prodejem energií v Kalifornii, a tedy by mohla tvořit skutečnou komunitu.

4.4.2 Závěrečné zhodnocení algoritmu

Podařilo se nám pomocí algoritmu odhalit překrývající se komunity v síti. Kvalitnější a reprezentativnější komunity jsme detekovali na rozšířeném grafu. Odhalili jsme dva základní druhy hranových komunit: první má základ v komunitách nalezených Lovaňským algoritmem a algoritmem SCAN, druhý popisuje vzájemnou komunikaci mezi dvojicemi komunitami prvního typu.

Pomocí překryvů můžeme nahlédnout do rozložení rolí v rámci větších komunit. Například růžová komunita na Obrázku 4.6 a její rozdělení na několik částí, kde každá zvlášť zajišťuje komunikaci s nějakou jinou komunitou. Nebo nám překryvy můžou ukázat důležitost vrcholů v komunitě, jako příklad v tomto případě může posloužit komunita B (žlutá) na Obrázku 4.6. Většinu komunikace komunity B s centrální růžovou komunitou zprostředkovávají uživatelé *allen* a *grigsby*.

Nepříjemnou vlastností naší implementace je velké množství detekovaných komunit a z toho plynoucí zkreslení sledovaných metrik, jako je například velikost komunit. Jedním z možných řešení tohoto problému by mohlo být zavedení omezeni na velikost komunit, které by se započítávali do konečných statistik.

4.5 Experimenty s dynamickým algoritmem DSCAN

Jak jsme popsali v Podkapitole 2.3.4, dynamickou síť lze reprezentovat několika způsoby. V případě dat z Enronu stav sítě v různých časových obdobích reprezentujeme pomocí výstřižků. Celkem jsme vytvořili 15 výstřižků, z nichž každý zastupuje jedno čtvrtletí od 4. čtvrtletí roku 1998 do poloviny roku 2002. Předpokládáme, že se struktura komunit bude v čase měnit.

Algoritmus aplikujeme pouze na rozšířený graf, protože změny probíhající v rozšířeném grafu zahrnují i změny v základní množině. Potvrzují to i experimenty na statických grafech – na grafu základních uživatelů i rozšířeném grafu měly komunity převážně stejné základy. Větší graf nám díky vyššímu počtu spojení mezi vrcholy navíc poskytne vyšší přesnost při detekci komunití.

DSCAN, jak napovídá i název, je rozšířením algoritmu SCAN. Rozdělení na prvním výstřižku najdeme pomocí SCAN algoritmu a v každém dalším výstřižku provádíme detekci komunit na základě rozdělení na komunity z výstřižku předchozího. Detekce spočívá v aktualizaci komunit a částí grafu, kde došlo ke změně hran nebo vrcholů (přidání, odebrání) pomocí aktualizační funkce.

Podobně jako SCAN má algoritmus dva parametry ε a μ, které se opět nastavují ručně. V našich experimentech budeme vycházet z optimálních hodnot nalezených pro SCAN aplikovaný na statický graf ($\varepsilon = 0.58$, $\mu = 4$ a $\varepsilon = 0.31$, $\mu = 3$).

Pro každý výstřižek budeme sledovat následující parametry:

- počet, velikost komunit a rozptyl velikostí;
- počet komponent souvislostí;
- hustota a průměr komunit;
- počet rozcestníků a odlehlých vrcholů.

V sledovaném období (od konce roku 1998 do poloviny roku 2002) došlo ve firmě Enron k odhalení podvodů a vyhlášení bankrotu, proto předpokládáme, že ke konci sledovaného období dojde k největším změnám ve struktuře komunit. Konkrétně se sníží velikost komunit spolu s ní i hustota a průměr, protože spojení mezi vrcholy bude čím dál méně. Na druhou stranu očekáváme zvýšení počtu komponent souvislostí a zvýšení počtu odlehlých vrcholů.

Bude nás tedy zajímat vývoj především během roku 2001, především jaký vliv mělo na komunity zahájení vyšetřování, období do žádosti o ochranu před věřiteli a následný úplný rozpad. Vývoj množství posílaných zpráv během celého sledovaného období je vidět na Obrázku 3.5. V polovině roku 2001, kdy začalo vyšetřování došlo k extrémnímu poklesu poslaných zpráv a ve čtvrtém čtvrtletí, kdy firma požádala o ochranu před věřiteli, došlo k náhlému vzrůstu a následnému rychlému poklesu množství poslaných zpráv. Kvůli popsaým výkyvům nemusí být interpretace získaných výsledků jednoznačná.

4.5.1 Výsledky získané na rozšířeném neorientovaném grafu

Potřebovali jsme podobně jako u algoritmu SCAN najít nejvíce vyhovující hodnoty parametrů ε a μ. Jako nejlepší hodnoty z algoritmu SCAN pro rozšířený graf nám vyšli dvojice parametrů ($\varepsilon = 0.58$, $\mu = 4$) a ($\varepsilon = 0.31$, $\mu = 3$). V dynamické verzi se nejlepších výsledků podařilo dosáhnout pro $\varepsilon = 4$ a hodnotu μ jsme museli snížit na 2. Udělali jsme to hlavně z toho důvodu, že některé výstřižky ze začátku sledovaného období nebo naopak z konce jsou velmi řídkými grafy, tudíž pokud bychom hranici podobnosti nastavili příliš vysoko, nenašel by algoritmus téměř žádné komunity.

Výsledky na jednotlivých výstřižcích jsme sledovali pomocí několika parametrů. Jako u všech předchozích algoritmů mezi nimi byly počet komunit, jejich průměrná velikost, rozptyl velikostí komunit, hustota a průměr komunit. Podobně jako u algoritmu SCAN jsme sledovali i počet odlehlých vrcholů a rozcestníků. Navíc jsme mezi sledované parametry přidali počet komponent souvislosti, protože předpokládáme, že v důsledku rozpadu komunit bude vznikat mnoho izolovaných vrcholů.

V Tabulce 4.15 jsou uvedeny výše zmiňované parametry pro všechny vygenerované výstřižky. Jak již bylo uvedeno, budeme sledovat především vývoj v roce 2001. Pro bližší analýzu jsme vybrali konkrétně první, třetí a čtvrté čtvrtletí roku 2001 a první čtvrtletí roku 2002 (v tabulce jsou řádky označeny $*$). Tyto výstřižky odpovídají extrémům ve výkyvech z grafu na Obrázku 3.5 na konci sledovaného období. Všechny výstřižky předtím nám poslouží k tomu, aby algoritmus postupně odhalil co nejpřesnější možnou strukturu komunit, která ve firmě existovala před sledovaným obdobím.

V Tabulce 4.15 je vidět, že od zahájení vyšetřování firmy Enron v polovině roku 2001 docházelo k postupnému zvyšování komponent souvislostí sítě a počtu odlehlých vrcholů. Lze to vysvětlit postupným oslabením spojení mezi zaměstnanci. Přímým důsledkem většího počtu odlehlých vrcholů je zmenšení průměrné velikosti komunit.

snapshot	NC	H	O	S	V	C	Diam	Dens
4_1998	0	0	244	0.00	0.00	225	0.00	0.00
1_1999	0	0	244	0.00	0.00	222	0.00	0.00
2_1999	1	0	209	35.00	0.00	207	5.00	0.10
3_1999	3	0	194	16.67	460.22	188	2.00	0.03
4_1999	14	0	159	6.07	125.49	143	1.14	0.15
1_2000	19	0	124	6.32	330.22	107	0.58	0.04
2_2000	24	1	91	6.33	395.14	76	0.71	0.12
3_2000	36	0	60	5.11	405.32	44	0.50	0.15
4_2000	31	1	48	6.29	595.11	28	0.55	0.13
* 1_2001	30	3	70	5.70	484.48	28	0.40	0.09
2_2001	17	0	37	12.18	1932.26	10	0.41	0.10
* 3_2001	17	3	46	11.47	1733.31	17	0.35	0.00
* 4_2001	24	0	27	9.04	1392.79	24	0.33	0.07
* 1_2002	17	1	75	9.88	1262.34	90	0.47	0.06
2_2002	21	1	116	6.05	452.43	236	0.10	0.00

Poznámka: 1) * jsou označeny výstřižky, na nichž blíže prozkoumáme vývoj struktury komunit.
NC–počet komunit, C–počet komponent souvislosti,
S–průměrná velikost komunity, V–rozptyl velikosti komunit,
O–počet odlehlých vrcholů, H–počet rozcestníků,
Diam–průměrný průměr komunity, Dens–průměrná hustota komunity.

Tabulka 4.15: Rozdělení vrcholů na komunity získaná algoritmem DSCAN s parametry $\varepsilon = 4$, $\mu = 2$ na výstřižcích rozšířeného grafu.

Vývoj největších komunit ve vybraných čtyřech výstřižcích lze sledovat na Obrázku 4.7. Některé komunity v průběhu zanikly (žlutá komunita H na grafu ze 3. čtvrtletí 2001), některé se rozpadly na menší (modré komunity D1 a D2 ze 4. čtvrtletí 2001, které vznikly z komunity D z 3. čtvrtletí 2001), jiné se naopak staly součástí větší komunity (světle modrá komunita F z 1. čtvrtletí 2001 se stala součásti modré komunity D v 3. čtvrtletí 2001).

Velikosti komunit se v průběhu sledovaného období znatelně měnily. Na přelomu let 2001 a 2002 lze pozorovat náhlý pokles velikostí všech komunit kromě D1 (modrá), nebo rozpad komunity E (tmavě růžová). Oranžová komunita C jako taková zanikla, ale několik jejích vrcholů (například uživatelé *allen* a *grigsby*) se staly součástí modré komunity D1.

Mezi označenými komunitami je také možné najít komunity zkoumané kvůli reprezentativnosti v algoritmech aplikovaných na statické grafy. Určíme je opět podle množin z Tabulky 4.4. Komunitě C z tabulky odpovídá fialová komunita G ve výstřižcích, komunitě B odpovídá na Obrázku 4.7 oranžová komunita C, komunitě A potom odpovídá tmavě růžová komunita E.

4.5.2 Závěrečné zhodnocení algoritmu

Algoritmus dokáže poskytnout náhled na vývoje sítě v čase a nalezené komunity mají základy odpovídající komunitám ve skutečném světě. Zjistili jsme, že události, které proběhly během sledovaného období ve firmě Enron, nezpůsobily výrazné změny ve struktuře komunit. Pozorované změny souvisely až se zánikem firmy. Mezi nejčastější změny můžeme zařadit rozpad komunit na menší celky, nebo zmenšení velikosti komunit, které někdy vedlo k připojení malé komunity k sousedící větší.

Nepříjemnou vlastností algoritmu je nutnost globálního nastavení parametrů ε a μ, které jsou poté používány napříč všemi výstřižky. Problém je, že výstřižky nemusí být všechny stejnorodé — některé grafy ze vzorku mohou být příliš řídké, jiné naopak příliš husté. Pro odhalení komunit v řídších grafech je třeba nastavit ε na nižší hodnotu, která ale v hustých grafech způsobí zařazení většiny vrcholů do jedné velké komunity. Vysoká hodnota ε naopak povede k nalezení komunit v hustých grafech, ale v řídkých grafech detekci komunit znemožní.

4.6 Vzájemné porovnání algoritmů

Při porovnání statických algoritmů budeme uvažovat pouze výsledky na neorientovaných a neohodnocených grafech, protože ani strukturální algoritmus SCAN, ani algoritmus klastrování pomocí hran nepodporují ohodnocené nebo orientované grafy. Při porovnání se omezíme na parametry z Tabulky 4.1. Přehled dosažených hodnot na rozděleních na komunity, které jsme u jednotlivých algoritmů působí jsme vybrali jako nejlepší, lze najít v Tabulce 4.16.

Nějvyšší modularity rozdělení se na obou grafech podařilo dosáhnout pomocí Lovaňského algoritmu. Je to předpokládaný výsledek, protože jako jediný hledá rozdělení na komunity s nejvyšší možnou modularitou. Hodnoty modularity pro zbylé dva algoritmy se od sebe příliš nelišily. Modularita se ale měnila mězi grafy, pro algoritmy SCAN a klastrování pomocí hran klesla v rozšířeném grafu o jednu desetinu.

Graf základních uživatelů						
algoritmus	M	NC	S	V	Diam	Dens
Lovaňský	**0.31**	15	9.87	147.32	1.53	0.72
SCAN	0.24	13	7.5	46.09	1.7	0.79
Klastrování pomocí hran	0.22	500	2.99	**8.19**	**1.3**	**0.83**
Rozšířený graf						
algoritmus	M	NC	S	V	Diam	Dens
Lovaňský	**0.29**	16	15.25	352.06	1.69	0.64
SCAN	0.13	8	11.38	78.48	2.13	0.72
Klastrování pomocí hran	0.13	1166	2.88	**6.58**	**1.29**	**0.85**

Poznámka: 1) tučně jsou zvýrazněny nejlepší hodnoty.

M–modularita, NC–počet komunit,

S–průměrná velikost komunity, V–rozptyl velikosti komunit,

Diam–průměrný průměr komunity, Dens–průměrná hustota komunity.

Tabulka 4.16: Nejlepší rozdělení získaná Lovaňským algoritmem, algoritmem SCAN a klastrováním pomocí hran na neorientovaném neohodnoceném rozšířeném grafu a grafu základních uživatelů.

Počet komunit nalezených klastrováním pomocí hran je pro oba grafy násobně větší, než u zbylých dvou algoritmů. Jak jsme popisovali v Podkapitole 4.4.1, vysoký počet nalezených komunit je nepříjemnou vlastností naši implementace. Vysoký počet komunit, jak bylo vysvětleno v Podkapitole 4.4.1 přímo ovlivňuje i nízké hodnoty průměrných velikostí a jejich rozptylu.

Pro Lovaňský algoritmus a SCAN jsou hodnoty, které se týkají počtu komunit a jejich velikostí porovnatelné, závisí vždy na konkrétním nastavení parametrů algoritmu SCAN. Algoritmy se výrazně liší v rozptylu velikostí detekovaných komunit. SCAN ho má výrazně nižší než Lovaňský algoritmus. Předpokládáme, že je to způsobené především tím, že Lovaňský algoritmus na datech z Enronu měl tendenci vytvářet jednu velkou centrální komunitu, která v některých případech obsahovala i třetinu všech vrcholů. U algoritmu SCAN sice dochází k obdobnému jevu pro nízké hodnoty ε, ale v rozdělení s nevyšší modularitou se nám tomu podařilo tvorbě dominantní komunity zabránit.

Nejkompaktnější komunity (s nejměnším průměrem a hustotou) dle Tabulky 4.16 jsme nalezli klastrováním pomocí hran. Je to ale opět způsobeno především velkým počtem malých detekovaných komunit. Pokud bychom porovnali pouze větší komunity, hustoty a průměry by byly porovnatelné se zbylými dvěma algoritmy.

Většina z větších detekovaných komunit nezávisle na tom, jaký algoritmus jsme použili, má základ ve skutečném světě. Téměř v každém rozdělení, s výjimkou několika nalezných algoritmem klastrování pomocí hran (Podkapitola 4.4.1), jsme byli schopní nalézt komunity, jejichž základy odpovídají množinám definovaným v Tabulce 4.4.

Nyní porovnáme výsledky algoritmu SCAN na statické síti s výsledky algoritmu DSCAN na dynamické. Většina z velkých komunit detekovaných na statických sítích se vyskytovala na téměř všech blíže zkoumaných výstřižcích. Mezi výstřižky z 4. čtvrtletí 2001 a 1. čtvrtletí 2002 jsme pozorovali největší změny, během kterých došlo například k zániku komunity C (Podkapitola 4.5.1).

Na závěr uvedeme porovnání času běhů algoritmů na jednotlivých grafech. Měření probíhalo na naši vlastní implementaci zkoumaných algoritmů (viz Příloha A.2). Nejpomalejším ze statických algoritmů vyšel algoritmus klastrování pomocí hran. Způsobené je to především velikostí dendrogramu (pro graf základních uživatelů měl 843 úrovní). Nejrychlejším z porovnání vychází algoritmus SCAN. Naměřené časy jsou k nahlédnutí v Tabulce 4.17.

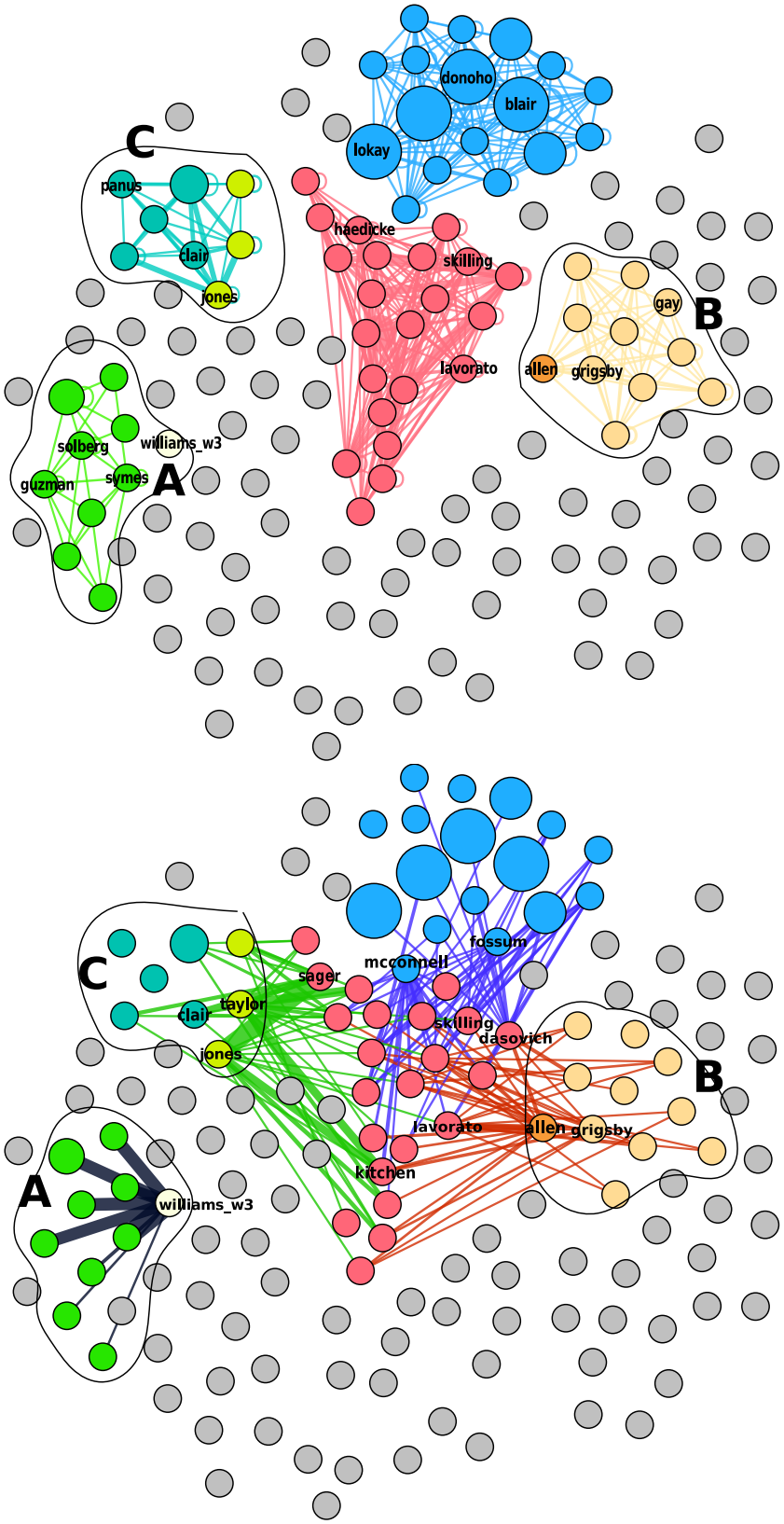
Kromě porovnání všech algoritmů na neorientovaných grafech, jsme provedli i porovnání běhů Lovaňského algoritmu na ohodnocených a orientovaných grafech, které lze najít v Tabulce 4.18. Jak lze vidět, detekce komunit na složitějších grafech trvala déle, než na základním neohodnoceném a neorientovaném grafu.

algoritmus	graf základních uživatelů	rozšířený graf
Lovaňský	32-35	43-46
SCAN	10-11	25-27
Klastrování pomocí hran	450-460	1515-1543
DSCAN	—	45800-41000

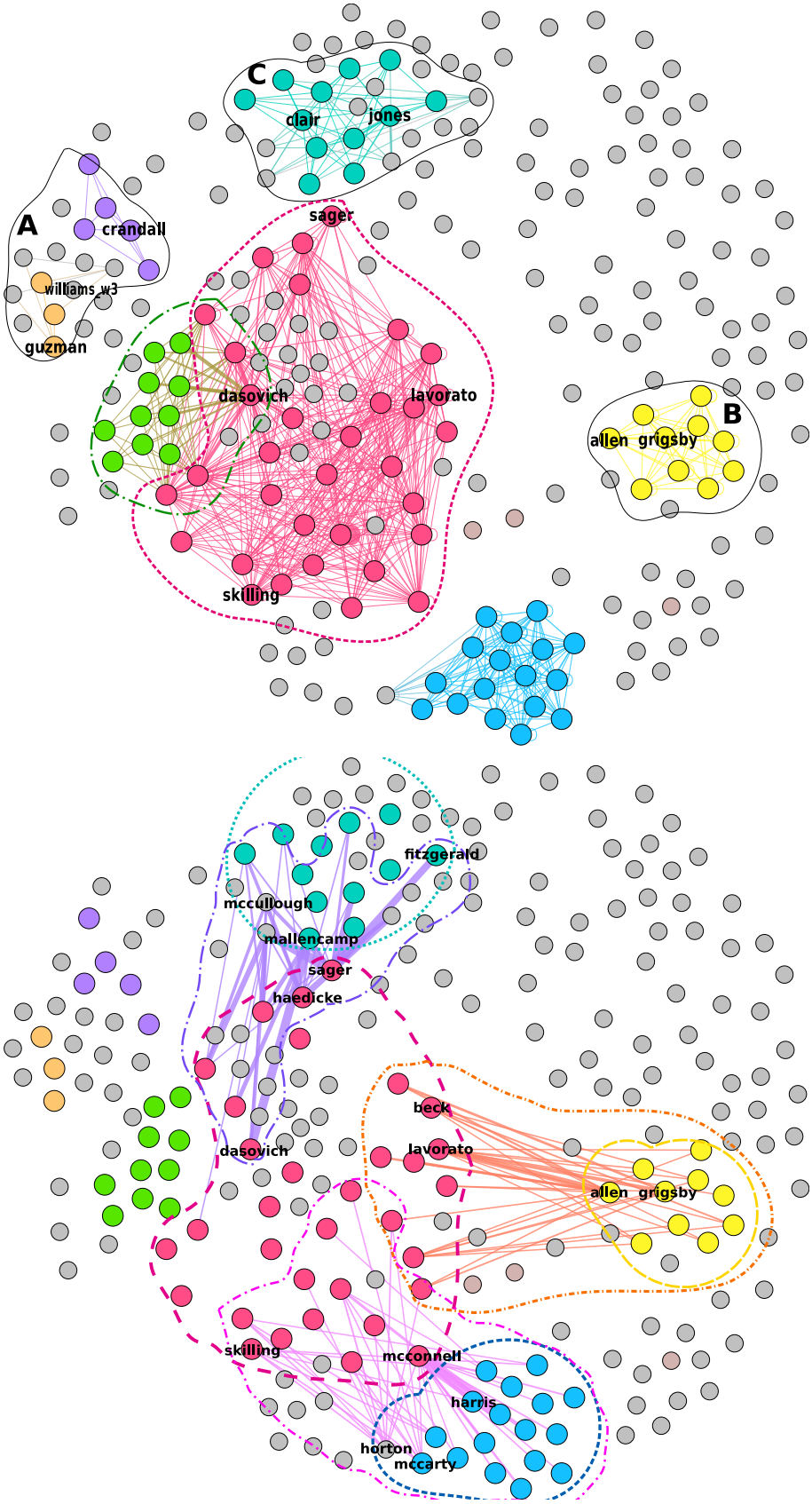
Tabulka 4.17: Porovnání rychlosti implementovaných algoritmů v milionech nanosekund (uvedeny jsou přibližné intervaly, v nichž se pohybovaly naměřené hodnoty).

typ grafu	graf základních uživatelů	rozšířený graf
základní	32-35	43-46
ohodnocený	36-38	50-54
orientovaný	30-33	55-65

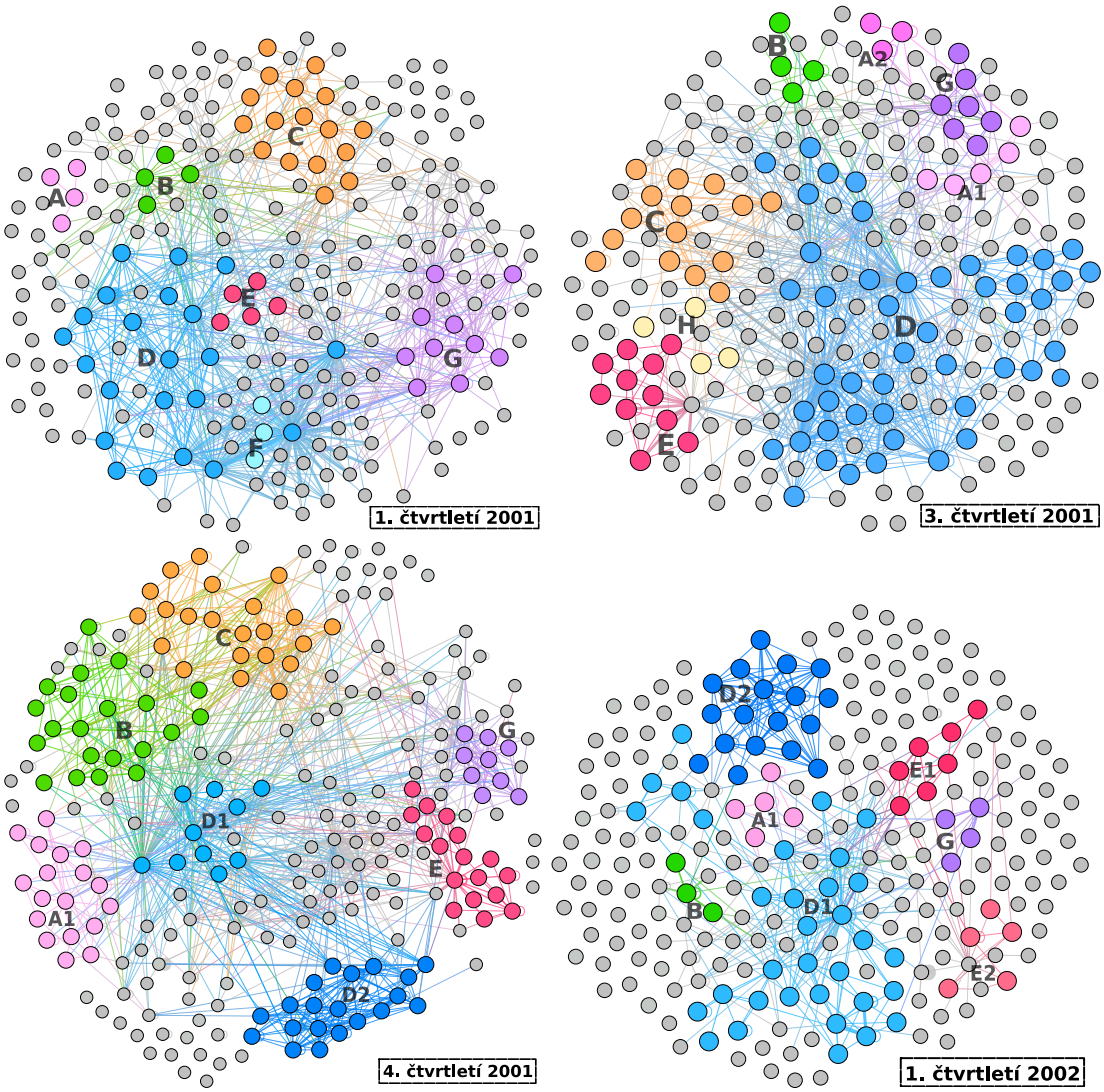
Tabulka 4.18: Porovnání rychlostí běhů Lovaňského algoritmu v milionech nanosekund na různých typech grafů (uvedeny jsou přibližné intervaly, v nichž se pohybovaly naměřené hodnoty).



Obrázek 4.5: Graf základních uživatelů s částmi rozdělení získanými klastrováním pomocí hran.
Horní graf obsahuje největší komunity, mezi kterými jsou dále vyznačeny komunity odpovídající komunitám A, B, C z předchozích algoritmů. Dolní zachycuje komunity spojující ty z horního grafu.



Obrázek 4.6: Rozšířený graf s částmi rozdělení získanými klastrováním pomocí hran.
Horní graf obsahuje největší komunity, mezi kterými jsou dále vyznačeny komunity odpovídající komunitám A, B, C z předchozích algoritmů. Dolní zachycuje komunity spojující ty z horního grafu.



Obrázek 4.7: Rozdělení získaná algoritmem DSCAN na výstřižcích z konce sledovaného období (1/2001, 3/2001, 4/2001, 1/2002).
Odpovídající si komunity jsou na všech výstřižcích označeny stejnou barvou.

Závěr

V této práci jsme se zabývali problematikou detekce komunit v sociálních sítích. Provedli jsme experimenty na síti tvořené e-mailovou komunikací zaměstnanců firmy Enron. V Kapitole 1 jsme definovali sociální sítě a komunity. Následně jsme uvedli různé způsoby detekce významných vrcholů a komunit a představili několik algoritmů, které jsme později použili k analýze struktury komunit na připravených datech (Kapitola 2). V Kapitole 3 jsme zanalyzovali dataset e-mailových zpráv firmy Enron a popsali grafy, na kterých jsme poté provedli experimenty, kterým se věnujeme ve zbytku práce (Kapitola 4).

Z algoritmů zmíněných v Podkapitole 2.2 jsme pro účely experimentů vybrali následující čtyři: Lovaňský algoritmus, algoritmus klastrování pomocí hran, SCAN a DSCAN. Vybrané algoritmy pokrývají všechny oblasti jmenované v Podkapitole 2.2: detekci hierarchické struktury, detekci překrývajících se komunit a detekci komunit v dynamických sítích. Jejich výběr jsme zdůvodnili v úvodu Podkapitoly 2.3.

Následně jsme představili dataset e-mailových zpráv mezi zaměstnanci Enronu z období od 30. října 1998 do 12. července 2002 (Kapitola 3). Zprávy jsme zanalyzovali a sestavili z nich několik grafů pro pozdější experimenty. Celkem jsme zkonstruovali pět grafů, z toho čtyři statické (Podkapitola 3.3) a jeden dynamický (Podkapitola 3.4).

Na zmíněné grafy jsme aplikovali vybrané čtyři algoritmy. Lovaňský algoritmus jsme použili pro odhalení hierarchické struktury a prozkoumání vlivu orientace a vah hran na výsledné rozdělení vrcholů na komunity. Z výsledků popsaných v úvodu sekce 4.2.1 jsme usoudili, že zkoumaná síť nemá výraznou hierarchickou strukturu. Co se týče vlivu vah na výslednou strukturu komunit, díky váženým hranám jsme byli schopní najít přesnější rozdělení s větším počtem hustších komunit (Podkapitoly 4.2.1 a 4.2.2). Orientace hran měla na strukturu podobný vliv jako započítávání vah. Rozdělení na grafech s orientovanými hranami obsahovala více menších komunit s nižším průměrem (Podkapitola 4.2.3). Z výše zmíněného tedy můžeme vyvodit, že v komunikačních sítích podobných síti zaměstnanců Enronu váha a orientace spojení hraje při detekci komunit velkou roli.

Algoritmus klastrování pomocí hran nám posloužil k odhalení překrývajících se komunit v síti. V průběhu experimentů jsme odhalili nepříjemnou vlastnost naší implementace algoritmu. Detekuje velké množství komunit, které se pohybuje ve stovkách, a většinu z nich tvoří malé komunity o 2-3 vrcholech. Kvůli tomu má algoritmus například v porovnání s ostatními nejnižší průměrnou velikost, rozptyl a průměr komunit (Podkapitola 4.6). I přes zmíněný nepříjemný fakt byl algoritmus při detekci překryvů užitečný. Odhalili jsme dva druhy komunit (Podkapitola 4.4.1), které se mezi sebou překrývají. První tvoří komunity podobné těm, které dokážou odhalit zbylé algoritmy (Lovaňský a SCAN), druhé reprezentují komunikaci mezi komunitami prvního druhu.

Pomocí algoritmu SCAN jsme odhalili několik rozcestníků, které byly předchozími dvěma algoritmy zařazeny do některé z větších komunit, kdežto v rozdělení nalezeném algoritmem SCAN je naopak spojují (Podkapitola 4.3.1). Nakonec pomocí algoritmu DSCAN se nám podařilo ukázat, že události, které vedly k zániku firmy, ovlivnily i strukturu komunit. Některé komunity zanikly, jiné se rozpadly

a některé se naopak staly součástí jiných (Podkapitola 4.5.1).

Každé získané rozdělení jsme navíc testovali na reprezentativnost pomocí hledání klíčových slov v předmětech posílaných zpráv. Pokusili jsme se použít i speciální grafy reprezentující komunikaci mezi uživateli v určitou denní dobu, ty se však neosvědčily. Klíčová slova jsme vyhledávali pro tři vybrané komunity a pro každou komunitu se nám podařilo odhalit skupinu uživatelů, která tvoří její jádro (Podkapitola 4.2.1). Tato jádra jsme použili k testování reprezentativnosti komunit napříč všemi algoritmy.

Na práci by se dalo navázat rozšířením implementace o další algoritmy, které by detekovaly například hlavní toky informací v síti. Jako další možnost se nabízí kvalitnější analýza reprezentativnosti komunit založená nejenom na hledání klíčových slov v předmětech zpráv, ale i v jejich obsahu. Kromě toho by rovněž mohlo být věnováno více prostoru praktické analýze významných vrcholů.

Seznam použité literatury

[1] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer, 2015.

[2] Albert-László Barabási and Márton Pósfai. *Network science*. Fourth Edition. Cambridge University Press, Cambridge, 2016.

[3] Andrea Landherr, B. Friedl, and Julia Heidemann. A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, 2(6):371–385, 2010.

[4] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[5] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási. Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297(5586):1551–1555, 2002.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[7] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. Random walks, markov processes and the multiscale modular organization of complex networks. *IEEE Transactions on Network Science and Engineering*, 1(2):76–90, 2014.

[8] M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[9] Balázs Adamcsek, Gergely Palla, Illés J. Farkas, Imre Derényi, and Tamás Vicsek. CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.

[10] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[11] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.

[12] Neo Christopher Chung, BłaŻej Miasojedow, Michał Startek, and Anna Gambin. Jaccard/tanimoto similarity test and estimation methods for biological presence-absence data. *BMC bioinformatics*, 20(Suppl 15):644, 2019.

[13] Rémy Cazabet, Giulio Rossetti, and Frederic Amblard. Dynamic Community Detection. In *Encyclopedia of Social Network Analysis and Mining*. Springer, New York, NY, 2017.

[14] N. Aston and W. Hu. Community detection in dynamic social networks. *Communications and Network*, 06:124–136, 2014.

[15] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: A structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 824–833, New York, NY, USA, 2007. Association for Computing Machinery.

[16] Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi. Qualitative comparison of community detection algorithms. *Digital Information and Communication Technology and Its Applications*, page 265–279, 2011.

[17] Natalie R. Smith, Paul N. Zivich, Leah M. Frerichs, James Moody, and Allison E. Aiello. A guide for choosing community detection algorithms in social network studies: The question alignment approach. *American Journal of Preventive Medicine*, 59(4):597–605, 2020.

[18] K. Mkhitaryan, J. Mothe, and M. Haroutunian. Detecting communities from networks: Comparison of algorithms on real and synthetic networks. 2019.

[19] Nicolas Dugué and Anthony Perez. Directed Louvain : maximizing modularity in directed networks. Research report, Université d'Orléans, 2015.

[20] Klimt Bryan and Yang Yiming. The enron corpus: A new dataset for email classification research:. In *Machine Learning: ECML 2004*, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[21] Julius Mansa) Investopedia (Troy Segal. Enron scandal, 2021.

[22] Roklen24. Krach enronu před 15 lety otřásl ekonomikou v usa, 2016.

[23] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle"in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011.

[24] Kevin Arvai. ikneed, 2020.

[25] Aric Hagberg, Pieter Swart, and Dan Schult. Networkx, network analysis in python, 2005.

[26] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. 2009.

[27] Thomas Aynaud. python-louvain, 2009-2018.

[28] The graphml file format, 2000.

[29] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

Seznam obrázků

Seznam tabulek

A. Implementace

S prací dodáváme zdrojový kód implementovaných algoritmů a program umožňující jejich použití z příkazové řádky. Kromě zdrojových kódů dodáváme i vygenerované grafy, na kterých jsme prováděli experimenty. V této příloze krátce popíšeme jak dodaný kód používat a uvedeme některé detaily implementace.

Kód byl implementován v jazyce Python, testován na operačním systému `Windows 10` a je závislý na `python 3.8` a `NetworkX 2.5.1`. Díky přenositelnosti `python 3.8` a `NetworkX 2.5.1` by ale neměl být problém spustit přiložené skripty na jiných platformách.

A.1 Uživatelská příručka

Přiložený .zip soubor obsahuje dvě složky `graphs`, obsahuje grafy vygenerovné z datasetu Enron v *.graphml* formátu, a `algorithms`, obsahuje zdrojové kódy a spustitelný skript `codea.py`. V této podkapitole popíšeme strukturu přikládaných souborů a způsoby použití přiložených programů.

A.1.1 Struktura

Grafy v `graphs` jsou rozdělené do tří složek: statické neorientované, statické orientované a dynamické neorientované. Složka s dynamickými grafy obsahuje všech 15 vygenerovaných výstřižků (Podkapitola 3.4).

Ve složce `algorithms` najdeme implementace jednotlivých algoritmů, metrik na analýzu získaných rozdělení, soubor s předměty zpráv na analýzu reprezentativnosti, grafy denních dob a testovací skript.

Výsledky se ukládají na stejném místě, kde je uložen vstupní graf (pokud není řečeno jinak), ve složce `results`, která je dále členěná na podsložky pro jednotlivé algoritmy. Pokud bychom jako příklad vzali podsložku `louvain`, bude obsahovat složky s jednotlivými běhy algoritmů pojmenované dle použitých parametrů, například `res_2_w_None_seed_None_dendr_True`. Složka reprezentující jeden běh algoritmu obsahuje graf s výsledným rozdělením (pokud jsme chtěli vygenerovat celý dendrogram, pak podsložky reprezentující jednotlivé úrovně), grafy s rozdělením aplikovaným na grafy denních dob (Podkapitola 3.5) a složku `statistics`, kde jsou uloženy metriky spočítané na nalezeném rozdělení na komunity.

Složka `statistics` obsahuje soubory `basic_properties.txt` (základní metriky, viz Podkapitola 4.1), `comm_centralities.txt` (centrality vrcholů pro jednotlivé komunity), `people_assigned_subjects.txt`, `subjects.txt` a `words_from_email_subjects.txt`, kde poslední tři obsahují data na analýzu reprezentativnosti.

A.1.2 Testovací framework

Každý z implementovaných algoritmů se dá použít zvlášť a zakomponovat libovolným způsobem do vlastního kódu. My ale představíme skript, pomocí kterého se dají detekovat komunity všemi představenými algoritmy a který umožňuje i analýzu výsledných rozdělení pomocí metrik představených v Podkapitole 4.1.

Testovací framework je tvořen jedním programem, `codea.py`, který se používá přes příkazový řádek. Pro použití je třeba mít instalovanou knihovnu `NetworkX` a program mít ve stejné složce jako zdrojové soubory s implementovanými algoritmy. Na vstupu jsou vyžadovány grafy v `graphML` [28] formátu. Výstupem jsou grafy s nalezeným rozdělením na komunity v `graphML` formátu.

Skript má jeden poziční parametr — soubor se vstupním grafem — a níže vypsané volitené parametry.

Parametry týkající se volby algoritmů:

- `-dynamic`: typ zvoleného algoritmu (`True`/`False`); předem nastaveno na `False`
- `-algorithm`: zvolený algoritmus; pro nedynamické lze vybírat z možností `all`/`linkc`/`louvain`/`scan`, pro dynamické `dscan`; počáteční hodnota `all`
- `-e`: parametr algoritmu SCAN; počáteční hodnota 0,6
- `-n`: parametr algoritmu SCAN; počáteční hodnota 2
- `-resolution`: parametr Lovaňského algoritmu; počáteční hodnota 1
- `-seed`: semínko pro randomizaci v Lovaňském algoritmu; počáteční hodnota `None`
- `-treshold`: parametr pro klastrování pomocí hran; počáteční hodnota 0
- `-weight_idf`: hranový atribut, který bude použit jako váha hran
- `-dendrogram`: výstupem budou rozdělení ze všech úrovní dendrogramu (pro Lovaňský algoritmus a klastrování pomocí hran); počáteční hodnota `False`
- `-results_directory`: složka, kam se mají uložit výsledná rozdělení a na nich spočítané metriky; složka musí existovat; pokud není zadána, ve složce, kde se nachází vstupní graf, se vytvoří složka `results`, kde v podsložce pojmenované dle použitého algoritmu budou uloženy výsledky

Parametry týkající se nastavení metrik:

- `-compute_metrics`: pro nalezená rozdělení spočítat i metriky (`True`/`False`)
- `-email_subjects`: soubor s předměty zpráv mezi uživateli firmy Enron; pokud zadán, skript vygeneruje textové soubory s údají, které lze použít na analýzu reprezentativity detekovaných komunit; možné použít pouze při analýze grafů vygenerovaných z datasetu Enron
- `-centr_to_g`: spočítané centrality budou uloženy jako atributy vrcholů v grafu s výsledným rozdělením (`True`/`False`)

Parametry na měření času [1]:

- `-alg_time`: změří běh algoritmu v nano sekundách, tj. nezapočítává dodatečnou analýzu získaného rozdělení, ani zápis grafu s výsledným rozdělením do souboru

[1] k měření se používá funkce `time.perf_counter_ns()`

- `-analysis_time`: změří celou analýzu grafu pomocí vybraného algoritmu, opět v nanosekundách, tj. zahrnuje nalezení rozdělení, změření metrik i zápis získaných dat do souborů

Příklad použití skriptu:

```
python3 codea.py \
../graphs/static_undirected/main_users_static_undirected.graphml \
--compute_metrics=True --centr_to_g=True \
    --email_subjects=mail_subjects.txt \
    --algorithm=louvain \
    --resolution=2 --weight_idf=weight --dendrogram=True \
    --results_directory=../graphs/static_undirected/main_results
```

A.2 Implementační detaily

Algoritmy ze sekce (2.3) jsou implementovány v jazyce `python` s podporou knihovny
`NetworkX` [25], která se specializuje na práci s grafy. Detaily implementace jednotlivých algoritmů jsou základním způsobem popsány dále v této příloze, pro bližší představu o implementaci může posloužit samotný okomentovaný kód. Vizualizaci všech výsledků v práci jsme provedli pomocí softwaru `Gephi` [26]. Údaje o délce běhů jednotlivých algoritmů lze nalézt v Podkapitole 4.6.

Při implementaci všech algoritmů jsme se inspirovali implementací Lovaňského algoritmu pro knihovnu `NetworkX` [27]. V každém algoritmu používáme jinak modifikovanou strukturu `GraphStatus`, ve které jsou uloženy předpočítané hodnoty, které se budou v průběhu algoritmu často používat. Podrobnosti každé struktury pro jednotlivé grafy lze nalézt níže v příloze.

Všechny grafy, které jsme vytvořili na základě dat z datasetu Enron (Kapitola 3), i grafy z výsledků experimentů, jsme ukládali v různými softwary a knihovnami široce podporovaném formátu `graphML` [28]. Formát umožňuje zadání grafů nejen se speciálně pojmenovanými vrcholy a hranami, ale i s pojmenovanými vrcholovými i hranovými atributy mnoha typů. Ukázku grafu v tomto formátu lze nalézt na obrázku (A.1).

Pro účely ukládání jsme vytvořili dvě funkce na přidávání získaných rozdělení jako atributů vrcholů, nebo hran v grafu.

První funkce `add_node_partition_as_attribute(graph, partition, partition_name, link=False)` na vstupu dostává graf, rozdělení vrcholů na komunity v podobě slovníku, název rozdělení, který se má použít jako jméno vrcholového atributu, a parametr `link` určije, jestli přidáváme rozdělení získané algoritmem klastrování pomocí hran.

Druhá funkce `add_edge_partition_as_attribute(graph, partition, partition_name)` přidává jako atribut rozdělení hran na komunity a používá se pouze u algoritmu klastrování pomocí hran.

Pro uložení grafu s přidanými atributy do souboru použijeme funkci knihovny `NetworkX` – `nx.write_graphml(graph, file_name)`.

```
<key id="w" for="edge" attr.name="weight" attr.type="int"/>
<graph edgedefault="directed">
<node id="haedicke"/>
<node id="williams_j"/>
<node id="bass"/>
…
<edge id="0" source="allen" target="steffes">
<data key="w">10</data>
</edge>
…
<edge id="1281" source="meyers" target="williams_w3">
<data key="w">7</data>
</edge>
```

Obrázek A.1: Ukázka orientovaného grafu s jedním hranovým atributem `weight`.

A.2.1 Lovaňský algoritmus

Implementovaný algoritmus se nachází ve složce `algorithms/louvain.py` a třída `GraphStatus` v souboru `algorithms/louvain_graph_status.py`.

Implementace obsahuje možnost nastavení rezolučního parametru, pomocí kterého jsme schopni snížit tendenci algoritmu slučovat při detekci menší komunity do větších. Implementace taktéž umožňuje randomizované (s volbou konkrétního semínka) procházení vrcholů v prvním kroku algoritmu (Sekce 2.3.1).

Pro zrychlení výpočtů využívám předpočítaných hodnot, které jsou udržovány ve třídě `GraphStatus`. Třída obsahuje položky vypsané v Tabulce A.1. `node_to_comm_dict` slouží na uložení aktuálního rozdělení vrcholů na komunity, `community_degrees` jsou stupně komunit, které se v průběhu algoritmu při přemísťování vrcholů mezi komunitami aktualizují.

název položky	komentář
node_to_comm_dict	slovník, kde klíče jsou vrcholy a hodnoty jim přiřazené komunity
total_weight	součet vah všech hran ve zkoumaném grafu
community_degrees	slovník, kde klíče jsou identifikátory komunit, hodnoty potom součty vah všech hran incidentních s vrcholy dané komunity
node_degrees	slovník, kde klíče jsou vrcholy, hodnoty stupně vrcholů v rámci grafu
weight_idf	jméno atributu hran, se kterým se počítá jako s váhou
resolution	parametr na škálování velikosti a počtu komunit
seed	semínko pro randomizované procházení vrcholů

Tabulka A.1: Přehled položek třídy GraphStatus pro neorientované grafy v Lovaňském algoritmu.

Pro orientované grafy je určena zvláštní třída `DIGraphStatus`, která má podobné položky jako `GraphStatus` jen uzpůsobené pro orientované grafy. Slovníky `community_degrees` a `node_degrees` byly každý narazeny dvojici slovníků pro vstupní a výstupní hrany.

Rozdělení na komunity je generováno funkcí `make_dendrogram(graph, resolution, weight_idf, seed=None)`, jejímž výstupem je dendrogram, který zachycuje komunity získané v každé iteraci algoritmu po skončení prvního kroku. Dendrogram je v tomto případě reprezentován speciálním způsobem jako seznam slovníků, kde množina hodnot slovníku na n pozici slouží jako množina klíčů pro slovník na $n + 1$ pozici. Pro získání rozdělení na komunity definovaného v n vrstvě dendrogramu slouží funkce `return_partition_from_level(n, dendrogram)`.

`def louvain(graph, resolution, weight_idf=None, seed=None, dendrogram=False)` je funkce, která je nabídnuta k použití uživateli. Jako parametry dostává graf (proměnná jednoho z grafových typů Graph nebo DiGraph definovaných v NetworkX), hodnotu rezolučního parametru, název atributu hran, který se má použít jako váha (nepovinné), semínko pro randomizaci (nepovinné). Poslední parametr určuje, jestli má algoritmus vrátit dendrogram se všemi nalezenými rozděleními, nebo pouze rozdělení s nevyšší modularitou, které se nachází v nejvyšší vrstvě dendrogramu.

A.2.2 Algoritmus klastrování pomocí hran

Implementovaný algoritmus i s příslušnou třídou `GraphStatus` se nachází ve složce `algorithms/link_clustering.py`.

Nejprve si představíme, jak vypadá třída `GraphStatus` pro tento algoritmus. Seznam všech položek se nachází v Tabulce A.2. Slouží nám především k uložení setříděného seznamu podobností (Vzorec 2.8) všech sousedících dvojic hran, kde jako sousední dvojici hran bereme takovou, která spolu sdílí právě jeden vrchol. Uržujeme si také aktuální rozdělení hran na komunity, množiny vrcholů příslušné k aktuálním hranovým komunitám a hustoty komunit.

Uživateli je nabídnuta funkce `link_clustering(graph, weight_idf=None, treshold=0, dendrogram=False)`, která na vstupu dostává neorientovaný graf, dále potom název atributu, který použijeme jako váhy hran, minimální podobnost hran, po které by dvě hrany měly být sloučeny do jedné komunity. Poslední parametr definuje, jestli uživatel na konci dostane úplný dendrogram, nebo pouze nejlepší rozdělení. Jako nejlepší rozdělení bereme to s největší hustotou (Vzorec 2.14).

Algortmus vrací rozdělení hran na komunity. Abychom z něj dokázali dostat rozdělení vrcholů na překrývající se komunity, použijeme funkci `return_node_partition(graph, edge_partition)`, která vráti dva slovníky. V obou jako klíče slouží vrcholy grafu. Hodnoty prvního jsou množiny identifikátorů všech komunit, do kterých patří příslušný vrchol. Jinými slovy obsahuje množinu všech komunit, do kterých patří některá z hran incidentních s daným vrcholem. Druhý slovník má ke každému vrcholu přiřazenou právě jednu hodnotu, a to komunitu, která je mezi hranami incidentními s daným vrcholem zastoupená nejvíce.

název položky	komentář
edge_to_comm_dict	slovník, kde klíče jsou hrany a hodnoty jim přiřazené komunity
comm_to_edge_dict	slovník, kde klíče jsou komunity a hodnoty odpovídající množiny hran
similarity_list	seznam dvojic hran setříděný dle jejich podobnosti
all_comm	slovník, kde klíče jsou všechny možné komunity vzniklé v průběhu algoritmu, hodnoty odpovídající množiny hran
comm_density	slovník, kde klíče jsou komunity a hodnoty jejich hustoty
comm_nodes	slovník, kde klíče jsou komunity a hodnoty množiny vrcholů incidentních s hranami, které patří do dané komunity
max_comm_id	maximální použitý identifikátor komunity
edge_number	počet všech hran v grafu

Tabulka A.2: Přehled položek třídy GraphStatus v algoritmu klastrování pomocí hran.

A.2.3 SCAN

Implementovaný algoritmus i s příslušnou třídou `GraphStatus` se nachází ve složce `algorithms/scan.py`.

Při implementaci jsme sledovali pseudokód 1 uvedený v druhé kapitole, jen s tím rozdílem, že jsme si předpočítali ε-okolí vrcholů a tyto hodnoty uchováváme ve třídě `GraphStatus`. Všechny položky této třídy jsou k nahlédnutí v Tabulce A.3. Kromě ε-okolí si udržujeme například aktuální rozdělení na komunity a hodnoty parametrů ε a μ.

název položky	komentář
node_to_comm_dict	slovník, kde klíče jsou vrcholy a hodnoty jim přiřazené komunity
e_neighborhoods	slovník, kde klíče jsou vrcholy a hodnoty jsou množiny všech sousedů daného vrcholu, které leží v jeho ε-okolí
n_param	hodnota μ
e_param	hodnota ε
max_index	maximální použitý identifikátor komunity

Tabulka A.3: Přehled položek třídy GraphStatus v algoritmu SCAN.

Uživateli je nabídnuta funkce `scan(G, e, n)`, která jako parametry dostává neorientovaný graf typu Graph z NetworkX, hodnotu parametru ε a μ. Funkce vrací slovník, kde jako klíče slouží vrcholy grafu a hodnoty jsou identifikátory přiřazených komunit, kterými může být buď unikátní číslo přiřazené komunitě, nebo klíčová slova *hub* (rozcestník) a *outlier* (odlehlý vrchol).

A.2.4 DSCAN

Implementovaný algoritmus se nachází ve složce `algorithms/dscan.py` a využívá třídu `GraphStatus` definovanou v souboru `algorithms/scan.py`.

Implementace DSCANu využívá třídu `GraphStatus` definovanou pro algoritmus SCAN (položky třídy k nahlédnutí v Tabulce A.3), kterou jsme rozšířili o dodatečnou funkci `update_neighborhoods(e, G)`. Slouží k aktualizaci ε-okolí vrcholů spojených hranou e při jejím přidání nebo odebrání z grafu.

Dále jsme také vytvořili speciální verzi funkce `scan(G, e, n)`, pojmenovanou `scan_for_dscan(G, e, n)`, která oproti původní verzi nevrací výsledné rozdělení vrcholů na komunity, ale poslední stav proměnné typu `GraphStatus`.

Uživateli je nabídnuta funkce `dscan(graph_stamps, e, n)`, která na vstupu dostává seznam neorientovaných grafů (výstřižků), hodnotu ε a μ. Vrací seznam rozdělení na komunity pro každý výstřižek ze seznamu `graph_stamps`. Každé rozdělení vrcholů na komunity je reprezentováno jako slovník, kde klíče jsou identifikátory vrcholů a hodnoty identifikátory přiřazených komunit.